GenRad

/APG

APG Equipped
ns

E

Instruction Manual

37363

Contents

Introduction – Section 1
Preparing Basic Inputs – Section 2
System Operation – Section 3
System Start-Up – Section 4
Preparing Simulation Data
(PREPARE Mode) – Section 5
DIAGNOSE Mode Use
(Board Testing) – Section 6
GENERATE Mode Use
(The CAPS Libraries) – Section 7
MAKE Mode – Appendix A
APG Use – Appendix B
LOG Mode – Appendix C
Nodal Verification – Appendix D
2225 Mode – Appendix E
Index

CAPS-X/APG

For CAPS/APG Equipped Test Systems

System X

Form 1796-0160-E

©GenRad 1984
Concord, Massachusetts, U.S.A. 01742
January, 1984

List of Effective Pages

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS 418 CONSISTING OF THE FOLLOWING:

Issue Revised Date January, 1984

Includes original (Jan., 1981) and
Change I (June, 1981) data.

Page No.	Issue
Title	January, 1984
A	January, 1984
DD-1 thru DD-5	January, 1984
v thru vi	January, 1984
1-1 thru 1-6	January, 1984
2-1 thru 2-36	January, 1984
3-1 thru 3-47	January, 1984
4-1 thru 4-5	January, 1984
5-1 thru 5-100	January, 1984
6-1 thru 6-64	January, 1984
7-1 thru 7-63	January, 1984
A-1 thru A-7	January, 1984
B-1 thru B-37	January, 1984
C-1 thru C-20	January, 1984
D-1 thru D-11	January, 1984
E-1 thru E-8	January, 1984
I-1 thru I-4	January, 1984

Difference Data

This publication (1796-0160) covers the use of CAPS X only. Customers with CAPS IX should refer to the 1796-0135 manual. CAPS X provides users new features and software improvements that can result in, among other enhancements, faster diagnosis probing with fewer points for the operator to probe. Additionally, other improvements result in more efficient software processing. The following enhancements, not available with earlier software releases, are included with this release:

NOTE

> The use of CAPS X software requires that a test system be configured to the System IX (Release 02) level. Use the CONVERT Mode described in your System Operation Manual to update CAPS/APG software from configurations earlier than CAPS/APG System IX (Release 02) to this level. Then, refer to the CAPS X Orientation Notes supplied with this release to update to CAPS X.

1. A conditional-connectivity capability has been added. This feature reduces the number of points that must be probed to diagnose the failure of a bussed node when 2 or more device outputs can drive the same node. The use of conditional connectivity is particularly suited for use with bus transceivers and devices with 3-state outputs. A conditional-connectivity file (.UC extension) can be established as described in para 5.2 that relates nodes to the state of the driving ability, i.e., a driving IC is probed only if it is drive enabled. Note also the Recent Additions section of this Difference Data for a description of the latest conditional-connectivity update.

2. A new Diagnostic Resolution Module (DRM) hardware option along with improved software is available. With CAPS X, either DRM I or the new DRM II can be used. DRM II features over DRM I include:

   40-pin clipping compared to 24 pins.

   Programmable, dual-threshold sensing with a selection of 2 sets of logic-family (A and B) thresholds compared to resistor-programmed single-threshold sensing levels.

   A microvoltmeter capable of measuring the microvoltage between either of 2 outside contacts in the probe and a center contact. The microvoltmeter probe can source/sink current with the third contact. This probe is used in conjunction with the FINDS software described in Section 6.

3. Automatic Fault Detection (AFD) allows a user to examine the fault detection of a test program. Prior to CAPS-X a test program could be separated into CAPS sections and simulated incrementally. Now, fault detection can be examined on trial sections as well. Additional macros have been developed as described in Appendix B to facilitate incremental AFD.

4. An enhancement to the specification of logic-level thresholds in the Setup Mode, and the specification of Diagnose submode is added for CAPS-X. Prior to this release, the same logic-level thresholds and diagnostic technique applied to the entire board, despite any use of multiple CAPS sections. Now, a file with a .DC extension can be created that allows the specification of thresholds and choice of Diagnose submode to be tailored to each CAPS section. The .DC file is described in Section 6.

5. A feature to permit the use of non-standard, though more familiar, user-assigned IC element names has been added. Non-standard IC names (those differing from library names) can be used in a .NM network if a cross-reference file called ALIAS.IC is first created and later processed during the Prepare N submode. During this submode the IC element library names are renamed to the more familiar user-assigned (alias) names. The use of the ALIAS.IC file is described in the PREPARE N Mode description in para 5.2.

6. CAPS-X software requires that the test system has 32K of core memory for proper operation. This requirement however, results in significantly faster probing diagnosis because of more efficient probe software execution.

7. The teletype bell rings to indicate another point is to be probed. With CAPS-X, the bell rings twice for a new point to probe, except when the next pin is located on the same IC as the last pin probed. The bell rings once when additional pins on the same IC are to be probed.

8. A Networking Mode has been added. This mode is used to specify board-test data-base files for transfer using the GenRad Network System. Refer to the Network System Operation Manual (Form No. 2291-0101) for a description of the NET Monitor Mode.

RECENT ADDITIONS

The following additions are available to current CAPS X users, and those with the special CAPS X update installed.*

An ICCOPY node that allows a user to copy a device from one library to another.

A change to conditional-connectivity (SJTRAN) that improves the preparation of the second section of the .UC file.

Each of these changes are described in the paragraphs that follow.

## ICCOPY

ICCOPY allows the user to copy a device (and all of its support ICs, if desired) from library "A" to library "B." This means that extensive regeneration of commonly used user elements will not be required for each new board being processed. Additional benefits accrue from being able to control which models are used by different programmers. The case of programmer "A" using source "A1" and programmer "B" using source "A2" to generate the "same" IC are gone. Also, it will now be possible to generate the SCL library once and copy only the elements needed into the user library. ICCOPY is a new mode accessed from the CAPS/APG monitor.

The ICCOPY monitor page is shown in Figure DD-1.

```
#                                                                  ICCOPY MODE


      File Usage

N=    6802               IC Element Name
FR=   DSK|:CAPSXA.CL     From Library
TO=   DSK:XYZ.CL         To Library
E=    YES                Also copy subelements required by this IC which are
                          contained in the same library
```

Figure DD-1. ICCOPY Monitor Page.

When an element is copied, the directory for that element name will show a C preceding the copy date as shown for the name AND10 below.

```
Name        Date            Name      Date          Name          Date

AM2569      12/10/80        ANCHOR     9/05/82      AND10       C 12/10/80
CD14517     12/10/80        CD4031    12/10/80      CD4031BC      12/10/80
CD4031BM    12/10/80        CD4076    12/10/80      CD4511        12/10/80
```

---

*Contact GenRad Applications Engineering, Concord, MA. if yor CAPS X disk does not contain the ICCOM or SJTRAN update and you desire it.

SJTRAN

Three additional capabilities have been added to conditional-connectivity (page 5-32) to address limitations in the translation process:

1. The ICTYPE from the entry in the second section of the .UC file is now optional. The translator will get the type used in the .SR file from the databases.

2. The pin number from the entry in the second section of the .UC file is also optional. The translator will provide all of the entries for that particular IC as mentioned in the first part of the .UC or CCDATA.LB.

3. All of the entries from the second section are optional by the use of the entry "ALLICS;." This will provide entries for all the ICs in the network which are wire-anded. This command may be used in addition to other commands as it will only add entries for IC pins not already done. It should be the first entry after the % separator in the second section.

This additional capability should ease the effort of creating the .UC. In addition, a new CCDATA.LB library is included which contains some additional elements and corrections not contained in previous libraries.

The file EXAMPLE.UC listed in Figure DD-2 shows the use of partial IC output pin specifications, and the ALLICS construct in Section 2 of a .UC file. There are two ways to generate a .UC file. The first way involves writing FILENAME.UC using one of the available editors, and making an entry for each pin of every tri-state device on the board. The second way allows the system to generate the .UC file. All that is necessary to have the system generate a .UC file is:

```
%                         /Section Separator
ALLICS;
```

The attribute ALLICS, when SJTRAN is run by using the "D" or "P" submodes of Prepare, enables SJTRAN to extract all of the wired-ands from the .ST file (the binary of the .SR file), and make a .UC file. Upon completion of this file, some additional work may be necessary. If an IC is not in the library CCDATA.LB, an entry "IC NOT FOUND" will be made in the .UC file. If the IC is not in the library, a user will have to make an entry in the first half of the .UC file (above the % separator).

Note that CCDATA.LB is a binary file. If a display of this file becomes necessary, the file CCDATA.HL can be used. This file can be used as a syntax guide for the first part of the .UC file (above the separator).

After editing the .UC file by removing any comments, any wired-and devices that are not tri-state devices, and the ALLICS attribute, it will be necessary to re-translate the file. This can be done using the "D" or "P" submodes of Prepare. If a successful translation occurs, an .I7 file will be created. This file will be used by the guided probe during Diagnosis if the "CC" option is enabled.

```
0001:  EXAMPLE.UC
0002:  8108RD (8-10,12-16)      IF (17 AND NOT 18);
0003:  2716BB (9-11,13-17)      IF (NOT 18 AND NOT 20);
0004:  25LS2520 (2,5,6,9,12,15,16,19) IF (NOT 20 AND NOT 21)
0005:  74LS244 (12,14,16,18)    IF (NOT 1);
0006:          (3,5,7,9)        IF (NOT 19);
0007:  74LS257 (4,7,9,12)       IF (NOT 15);
0008:  AM9519BB (4-11)          IF (NOT 1 AND NOT 27 AND NOT 3 AND 2 AND 26) OR
0009:                            (NOT 1 AND NOT 3 AND 27 AND 2 AND 26) OR
0010:                            (1 AND NOT 26);
0011:  SMC5025 (10-17,24-31)   IF (NOT 18 AND 25);
0012:  IN8155 (1,2,12-19,21-39) IF (NOT 8 AND NOT 9 AND 4 AND 11 AND 7);
0013:  74LS04 (2)IF(1),(4)IF(3),(6)IF(5),(8)IF(9),(10)IF(11),(12)IF(13);
0014:  75276 (16,15,6,5)IF(11 OR 1);
0015:  %
0016:  ALLICS;
0017:  I20.2   74LS245;
0018:  I20.3;
*/*    I20.3   74LS245;
0019:  I21;
*/*    I21.2   74LS245;
*/*    I21.3   74LS245;
*/*    I21.4   74LS245;
*/*    I21.5   74LS245;
*/*    I21.6   74LS245;
*/*    I21.7   74LS245;
*/*    I21.8   74LS245;
*/*    I21.9   74LS245;
*/*    I21.11  74LS245;
*/*    I21.12  74LS245;
*/*    I21.13  74LS245;
*/*    I21.14  74LS245;
*/*    I21.15  74LS245;
*/*    I21.16  74LS245;
*/*    I21.17  74LS245;
*/*    I21.18  74LS245;
*/*    I20.4   74LS245;
*/*    I20.5   74LS245;
*/*    I20.6   74LS245;
*/*    I20.7   74LS245;
*/*    I20.8   74LS245;
*/*    I20.9   74LS245;
*/*    I20.11  74LS245;
*/*    I20.12  74LS245;
*/*    I20.13  74LS245;
*/*    I20.14  74LS245;
*/*    I20.15  74LS245;
*/*    I20.16  74LS245;
*/*    I20.17  74LS245;
*/*    I20.18  74LS245;
*/*    I23.12  74LS244;
*/*    I23.14  74LS244;
*/*    I23.16  74LS244;
*/*    I23.18  74LS244;
*/*    I23.3   74LS244;
*/*    I23.5   74LS244;
```

Figure DD-2. Example .UC File.

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_06653

Table of Contents

Para Page

SECTION 1 - INTRODUCTION

1.1 PURPOSE. 1-1
1.2 DESCRIPTION. 1-2
1.3 MAKING AN INITIAL CAPS RUN 1-4
1.4 CARE AND STORAGE OF DISK CARTRIDGES. 1-5

SECTION 2 - PREPARING BASIC INPUTS

2.1 INTRODUCTION 2-1
2.2 DEVICE ADAPTOR AND PROBE/CLIP PREPARATION. 2-1
2.3 CODING OF THE NETWORK DESCRIPTION. . 2-2
2.4 PREPARING A TEST PROGRAM 2-31
2.5 CODING BLACKBOX MODELS 2-31
2.6 DRM SENSOR THRESHOLD SELECTION . . . 2-33

SECTION 3 - SYSTEM OPERATION

3.1 INTRODUCTION 3-1
3.2 DEC OS/8 OPERATING SYSTEM. 3-5
3.3 GenRad TEST, CAPS, AND APG MONITORS. 3-29

SECTION 4 - SYSTEM START-UP

4.1 INTRODUCTION 4-1
4.2 SYSTEM SOFTWARE. 4-1
4.3 STARTING PDP8/A SYSTEMS. 4-2
4.4 STARTING PDP8/E SYSTEMS WITH DEC OR PLESSEY DISK CONTROLLERS. 4-3
4.5 STARTING PDP8/E SYSTEMS WITH DSD DISK CONTROLLERS 4-4
4.6 SYSTEM SHUT-DOWN 4-5

SECTION 5 - PREPARE MODE

5.1 INTRODUCTION 5-1
5.2 PREPARE MODE SOFTWARE. 5-1
5.3 PREPARE MODE OPTIONS 5-48
5.4 PREPARE MODE DATA OUTPUTS. 5-63
5.5 PERFORMING A TYPICAL RUN 5-78
5.6 FILE MANAGEMENT. 5-82
5.7 SPECIAL FEATURES 5-99

SECTION 6 - DIAGNOSE MODE

6.1 INTRODUCTION 6-1
6.2 DIAGNOSTIC TECHNIQUES AVAILABLE. . . 6-1
6.3 DIAGNOSE MODE OPTIONS. 6-11
6.4 DIAGNOSTIC ERROR MESSAGE FORMATS . . 6-22
6.5 COMPUTER-GUIDED PROBE//CLIP TESTING. 6-32
6.6 FINDS TESTING. 6-47
6.7 AFL TESTING. 6-57
6.8 BOARD TESTING IN MANUAL SUB-MODE . . 6-64

Para Page

SECTION 7 - GENERATE MODE

7.1 INTRODUCTION 7-1
7.2 IC MODELING - GENERAL PRINCIPLES . . 7-2
7.3 PREPARING IC MODELS. 7-2
7.4 ADDING IC MODELS TO THE USER OR UPDATE LIBRARY 7-9
7.5 LIBRARY UTILITY PROGRAMS 7-11
7.6 IC MODELING CONSIDERATIONS 7-16
7.7 ROM GENERATION 7-21
7.8 BOOT ELEMENTS IN THE IC LIBRARY. . . 7-28

APPENDIX A - MAKE MODE

A.1 INTRODUCTION A-1
A.2 PREPARING CAPS DATA BASE FILES . . . A-1
A.3 USING MAKE MODE. A-2
A.4 USING FLFORM PROGRAM A-5

APPENDIX B - APG USE

B.1 INTRODUCTION B-1
B.2 INCREMENTAL SIMULATION B-1
B.3 E/T/C MODE B-20

APPENDIX C - LOG MODE

C.1 INTRODUCTION C-1
C.2 PREREQUISITES. C-1
C.3 ACCESSING LOGGING MODE C-2
C.4 OPTION PAGES C-2
C.5 FILE INITIALIZATION. C-6
C.6 LOG DATA COLLECTION. C-7
C.7 LOG FILE STATUS DISPLAY. C-7
C.8 LOG FILE REPORT GENERATOR. C-8
C.9 CAUTIONS C-19
C.10 LOG MODE PRINTOUT. C-20

APPENDIX D - NODAL VERIFICATION

D.1 INTRODUCTION D-1
D.2 USING NODALV D-3
D.3 NVTRAN D-8
D.4 ERROR MESSAGES D-9
D.5 SHORTS CHECK MODE. D-10

APPENDIX E - 2225 MODE

E.1 INTRODUCTION E-1
E.2 REQUISITES E-1
E.3 2225 MODE OPTIONS. E-3
E.4 IMAGE TRANSLATOR ERRORS. E-5
E.5 IMAGE TRANSLATOR WARNINGS. E-5
E.6 TEST PROGRAM TRANSLATOR ERRORS . . . E-6
E.7 TEST PROGRAM TRANSLATOR WARNINGS AND MESSAGES E-6
E.8 PDP-8 TO 2225 LINK E-7
E.9 KL8JA CONSIDERATIONS E-8

The material in this manual is for informational purposes only and is subject to change without notice. GenRad, Inc. assumes no responsibility for any errors or for consequential damages that may result from the use or misinterpretation of any of the procedures in this publication.

NEW GR FAULT TRACER AVAILABLE

The GenRad 2220 Bug Hound is a completely self-contained instrument designed to take over from the CAPS FINDS software. With the CAPS FINDS diagnostic message, which gives the electrical location of the fault, a technician can select one of the Bug Hound's several operating modes and quickly track down the physical location of the fault.

The Bug Hound has a unique new phase-sensitive current-tracing probe* that makes it easy to trace the correct track even when many narrow tracks are running very close together. In addition, it has a dc tracing capability, an ac current source, and a dc current source.

Major features and benefits include:

- Locates "shorts" and "opens".
- Locates the bad IC on a node
- Locates shorts between power and ground Tracks – including shorted capacitors.
- Works for all logic families.
- Requires no power on the board – therefore it does not tie up the Test System or other instrumentation such as power supplies or scopes.
- Easy to use – requires no "sensitivity" or "calibration" adjustments.

---

*Patent Pending

GenRad
2220 Bug Hound

Introduction – Section 1

1.1 PURPOSE

This manual contains operating instructions for CAPS (Computer Aided Programming Software) and Automated Program Generation (APG) equipped test systems. It includes all procedures necessary to prepare the basic CAPS/APG inputs required, and then operate CAPS or APG as it simulates logic, tests boards, and automatically isolates faults.

The manual is divided into the following sections:

Section 1 - Introduction
Section 2 - Preparing Basic Inputs
Section 3 - System Operation
Section 4 - System Start-up
Section 5 - Preparing Simulation Data (PREPARE Mode)
Section 6 - DIAGNOSE Mode Use (Board Testing)
Section 7 - GENERATE Mode Use (The CAPS Libraries)
Appendix A - Making 1795 Disks/Diskettes (MAKE Mode)
Appendix B - APG Use
Appendix C - LOG Mode
Appendix D - Nodal Verification (NODALV Mode)
Appendix E - 2225 Mode
Index

Section 1 includes introductory material, a brief overview of CAPS and APG, major features of both, and how to use them effectively. Some basic procedures covering the care and use of disk cartridges are also included.

Section 2 describes the basic inputs that are needed to run CAPS or APG and how to prepare them. Preparation may be done "off-line" without the aid of a test system.

Section 3 describes the GenRad system software and contains some basic examples of how to use it effectively. Methods of accessing major modes and programs through use of the GenRad monitors (CAPS and APG) and methods of accessing programs from the DEC OS/8* keyboard monitor are covered.

Section 4 includes procedures for starting-up and re-starting CAPS or APG equipped test systems. General precautions that should be followed during start-up are also covered.

Section 5 describes the optional commands, procedures, and software used during a PREPARE Mode run to simulate a known good board, model selected faults, and prepare the data base files needed for board testing in the DIAGNOSE Mode.

Section 6 describes the optional commands, procedures, and software used to test logic boards and locate failures in the DIAGNOSE Mode.

---

*Registered Trademark of Digital Equipment Corp., Maynard, MA.

Section 7 describes the optional commands, procedures and software used in the GENERATE Mode to add or change elements in the CAPS user (USERLB) and update (UPLIB9) libraries. It also lists and explains a series of special purpose elements in the two CAPS IC Libraries (CAPS9 and CAPS9A) that can be used as IC building blocks when constructing other elements for entry in the user library.

Appendix A describes the procedures necessary to make disks or diskettes for use with 1795/1799 test systems. It includes the steps necessary to create the data base files required for 1795/1799 testing and then copy the files onto a disk or floppy diskette.

Appendix B covers the use of the Automated Program Generation Package (APG). Included within APG is Incremental Simulation and use of the Equation Generation, Tracking, and Connectivity (E/T/C) Mode.

Appendix C describes LOG Mode. This mode may be used to log CAPS data resulting from a DIAGNOSE Mode run in a file. Flexible control allows the user to log specific CAPS data (such as a compact form of the Probe History, First Failing Test, etc.) and then output the accumulated data selectively.

Appendix D describes Nodal Verification (NODALV Mode.) This mode is used to automatically prepare Pulse Discrimination (.UD) files for use with the GenRad pulse catching probe in DIAGNOSE Mode.

Appendix E describes the 2225 Mode. This mode is used to Translate CAPS network descriptions and test programs into a form usable by the 2225.

The Index at the rear of the manual may prove extremely useful when searching for specific subjects within the publication. The index is arranged in alphabetical order for ease of reference.

1.2 DESCRIPTION

1.2.1 General

Both CAPS and APG are software options for use with a variety of GenRad disk based test systems. All software modules are recorded on a CAPS or APG system cartridge that is installed in the top drive of multiple disk systems or the only drive of single disk systems. Both CAPS and APG operate in the following three basic modes:

1. PREPARE Mode. This mode is used to simulate the operation of a known good logic board without the aid of any physical hardware whatsoever (an actual known good board is not required). Once the good board simulation has been completed, selected faults may also be modeled by simulation.

2. DIAGNOSE Mode. This mode is used to test actual logic devices and to automatically isolate the vast majority of circuit failures that can occur in a production environment. A computer-guided diagnostic probe/clip is also provided for analysis and detection of extremely complex circuit failures. In addition, an optional Diagnostic Resolution Module (DRM) working in conjunction with Fault Identifying Nodal Diagnostic Software (FINDS) enables nodal failure diagnoses to be further refined, when desired.

3. GENERATE Mode. This mode is used to enter and modify simulated elements in the CAPS User (USERLB) and Update (UPLIBX) libraries. Only ICs included in one of the CAPS libraries can be modeled during a simulation run. This mode allows new ICs to be entered into the user library (or existing ICs changed) in a rapid and efficient manner.

Each of these three basic modes are applicable to both CAPS and APG and are described further in Para. 1.2.2. When APG software is supplied, the following additional modes are included.

Incremental Simulation

This involves a new method of test program generation that allows a user to build a program incrementally, without requiring re-simulation of previously processed segments of the program. The test program may be edited, simulated, analyzed, and conditionally accepted or rejected "on-line" through use of the I sub-mode of PREPARE furnished with APG equipped test systems. Incremental Simulation is described in Appendix B.

E/T/C Mode

The E/T/C Mode is the second part of APG described in Appendix B. It provides the tools necessary to help a user "debug" test programs by pinpointing the causes for such problems as "X" (or indeterminate states), pulses, and transitions occurring on active pins at any test step. The mode may also be used to obtain a better understanding of the circuit being simulated so that improvements can be added to the test program used to drive the simulated model.

1.2.2 PREPARE Mode

During a PREPARE Mode run, the CAPS software simulates a known good version of the actual board to be tested without the aid of any physical board. As a result, an engineer can analyze and verify circuit operation before a prototype board is even available if he so desires. Since CAPS does not require a physical board in order to determine correct output states, it can also model unknown "X" states (that is, while an actual board's outputs will either be in a known 1 or 0 state, the state selected may be in error at a given time, due to critical race conditions, oscillations, improper initialization, etc. The simulated model will flag problems of this nature by listing the output states as "X" or unknown).

The only inputs required in order to simulate a known good board are a network description and a basic test program. The network description provides CAPS with a complete description of the logic device to be simulated and tested. The basic test program provides the Input and Output patterns necessary to test a logic circuit. It is prepared in the GenRad "high-level" test language and may consist of Input patterns only. Output patterns are generated automatically (digitally autoprogrammed) during the good board simulation run without the aid of a known good board.

After generating a model of a known good board, and providing a digitally autoprogrammed test program that includes output states, a fault simulation run can begin. During the fault simulation run, a series of optional faults may be modeled (such as IC Inputs and Outputs stuck-at-zero and stuck-at-one, shorts to ground or V_{cc} tracks, open connections to IC pins, power loss to an IC, shorts between adjacent IC pins, and more). Optional selection of faults to be modeled makes it possible to select only those failures that are most likely to occur in any given circuit while ignoring others, or model all possible failures. A complete description of all faults that can be modeled is contained in Section 5. As the faults selected are modeled, and faulty boards are simulated, the faulty outputs obtained are compared against the known good board outputs at each test step to evaluate the efficiency of the basic test program (how good is it in detecting faults?). Data outputs identify both detected and undetected faults so that the basic test program can be easily improved. Diagnostic files are also created and are used later for analysis of circuit failures during logic device testing in the DIAGNOSE Mode.

The PREPARE Mode then, can be used to simulate a known good board, model selected faults, and finally provide both the diagnostic data needed for logic device testing and analysis in the DIAGNOSE Mode, and data that can be used to evaluate the efficiency of the basic test program. As a direct result of the good board simulation, other data outputs are also provided that can be used for a complete evaluation of the circuit being simulated. For example, a complete listing of the simulation run that includes the state of each primary input, primary output, and device output pin at each test, or a complete wire-wrap list that includes board input and output connections to circuit ICs and then lists connections from circuit ICs may be provided. A complete description of all PREPARE Mode optional commands and data outputs available is contained in Section 5.

1.2.3 DIAGNOSE Mode

The DIAGNOSE Mode allows rapid testing and anlaysis of actual logic circuits. When board failures occur, the CAPS software can automatically analyze and diagnose the majority of circuit failures that may be encountered in a production environment. A computer-guided probe/clip is also provided for analysis of any complex circuit failures that may occur. Probe instructions inform the user of those points that are to be probed. Probe diagnostic messages automatically identify failed components. The FINDS software package is also available to further diagnose nodal failures, when required. A complete description of all DIAGNOSE Mode optional commands and diagnostic messages is contained in Section 6.

1.2.4 GENERATE Mode

The GENERATE Mode options are used to enter simulated elements in the CAPS User and update libraries. A CAPS IC Library is also furnished and has available a series of complete IC models (7400,7440,7496,etc.) extracted from a variety of IC manufacturers catalogs, and a series of special purpose elements that can be used as "building blocks" to construct other, more complex IC models that can be placed in the user library. The GENERATE Mode options allow new ICs to be entered into the user library and existing ICs in the user library to be modified in a rapid and efficient manner. A complete description of all GENERATE Mode optional commands, data outputs, and "building blocks" is contained in Section 7.

1.3 MAKING AN INITIAL CAPS RUN

This manual is divided into sequential sections that may be followed whenever a new circuit is being made ready for testing and fault determination in the DIAGNOSE Mode. Basically, the procedures involve:

1. Preparing the initial CAPS Inputs. Prior to making an initial CAPS run for a new board, it will be necessary to select and wire the CAPS probe pin (if the X-1 probe is to be used), and prepare both a network description and basic test program. All of these procedures can be accomplished "off-line" without the aid of a test system. Details are contained in Section 2.

2. Starting the test system and entering the keyboard monitor (refer to sections 3 and 4).

3. Creating the network description (.SR) file and basic test program (.TP) file with EDIT, and then checking each file for syntax and other errors. Next, a series of simulation runs can be made and the basic test program improved until it has sufficient fault detection diagnostic resolution for the board to be tested. Details are contained in Section 5.

4. Once a satisfactory simulation run has been completed in the PREPARE Mode, the DIAGNOSE Mode can be entered for device testing and analysis. Details are contained in Section 6.

1.4 CARE AND STORAGE OF DISK CARTRIDGES

Proper care and storage of disk cartridges can prevent loss of data. The following precautions should be observed:

1. Store cartridges in a closed cabinet to eliminate contamination by foreign matter.

2. Storage facility environmental considerations are important. The following ambient temperature and relative humidity ranges should be maintained whenever possible:

   60°F to 90°F (16°C to 32°C) with 40% to 60%
   relative humidity (without condensation).

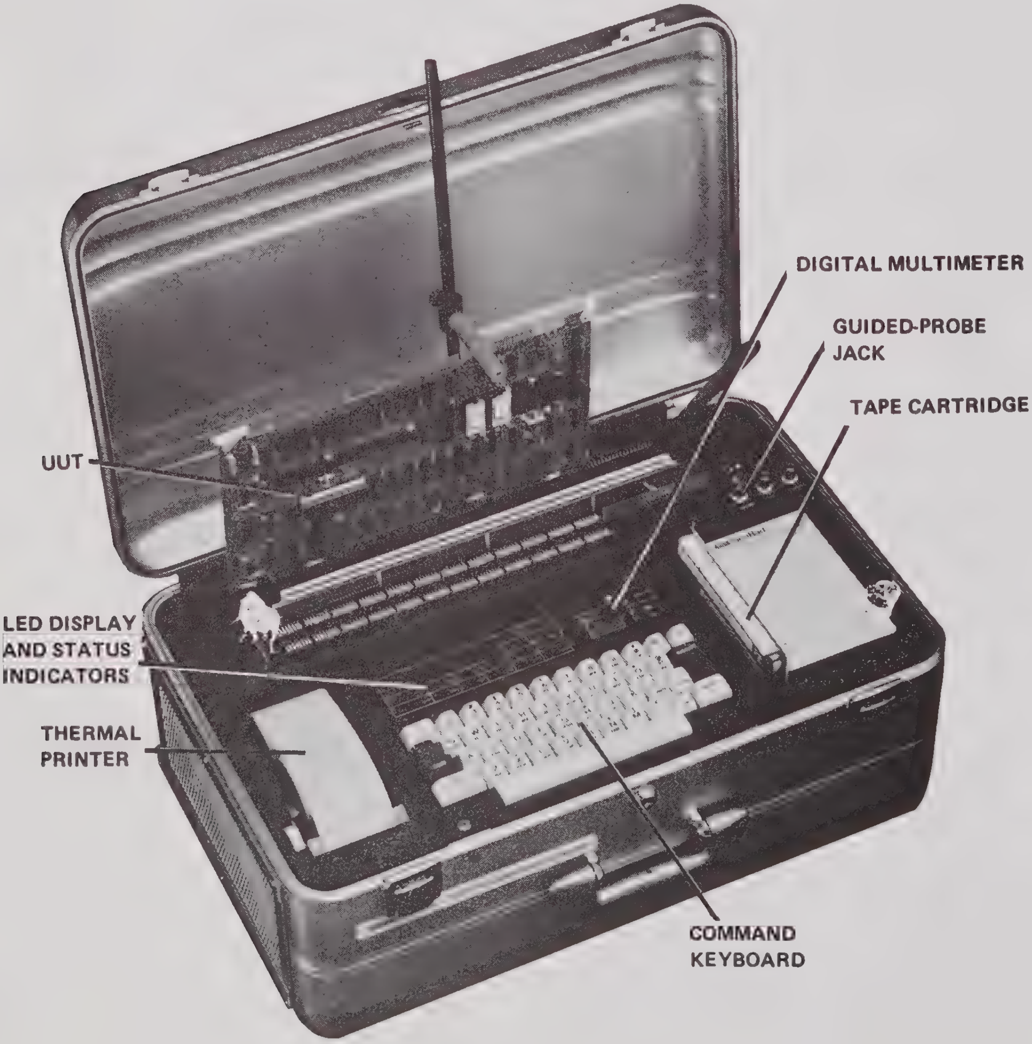
3. Cartridges exposed to conditions beyond the limits specified in step 2 for long periods should be reconditioned before use. Reconditioning is accomplished by placing the cartridge in the specified environment for the same length of time that it has been removed before use (up to a maximum of 24 hours).

4. Inspect cartridges to ensure that dust covers are closed when they are not installed in disk drives.

5. To avoid loss of data, always keep cartridges away from any source of electromagnetic radiations.

NEW GenRad PORTABLE SERVICE TESTER AVAILABLE

The GenRad 2225 Portable Service Tester is a completely self-contained instrument designed for fast diagnosis and repair of faulty complex digital PC boards in the field, thus reducing large spares inventories. It is the first portable tester with automatic guided-probe fault isolation, interactive programmable power supplies, driver/sensors, and most of the features found only in more expensive production-oriented test systems. The unit may be interconnected directly to GenRad CAPS/APG equipped test systems so that test programs and probe data prepared for the larger test systems can be easily translated into files that are suitable for use on the 2225.


DIGITAL MULTIMETER
GUIDED-PROBE JACK
TAPE CARTRIDGE
UUT
LED DISPLAY AND STATUS INDICATORS
THERMAL PRINTER
COMMAND KEYBOARD

Preparing Basic Inputs – Section 2

2.1 INTRODUCTION

This section lists the procedures necessary to prepare basic CAPS/APG Inputs for use. The following basic inputs and selections are required:

1. A wired device adaptor. The device adaptor interconnects the Unit Under Test (UUT) with the test system driver/sensor pins and device power supplies. Details are contained in Para.2.2.1.

2. A probe or clip to be used as the computer-guided (smart) probe or clip. Either a X-1 passive probe (that can record good and bad logic levels) or a GenRad pulse-catching probe (that can also record illegal or missing pulses, pulse transitions, and oscillations) may be used as the smart probe. In addition, a 16 (or 24) pin IC clip can be specified in place of the X-1 passive probe but does not provide any pulse-catching capability. If the X-1 passive probe is to be used as a smart probe, a dedicated driver/sensor pin must also be selected at this time. Details are contained in Para. 2.2.2.

3. A network description (.SR file). The network description provides CAPS or APG with a complete description of the logic device to be simulated and tested.

4. A test program (.TP file). The test program provides the Input and Output patterns necessary to test a logic circuit. It is prepared in our GR "high level" test language and may consist of Input patterns only. Output patterns are generated automatically by CAPS/APG (digitally autoprogrammed) during a simulation run. Details are contained in Para. 2.4.

5. The proper DRMI Resistor Values. The original Diagnostic Resolution Module (DRMI) has a series of selectable resistor voltage divider networks that determine the DRM sensor thresholds for each logic family encountered during test. Four thresholds are pre-set by GenRad to cover TTL, ECL, 3-state logic, and provide a check of clip connections. Four thresholds are customer selected. Methods of selecting and installing resistors to establish or change the threshold values are described in Para. 2.6. For systems equipped with the DRMII, the thresholds are software-selectable as described in para 2.6

2.2 DEVICE ADAPTOR AND PROBE/CLIP PREPARATION

2.2.1 Device Adaptor Preparation

Device adaptors are supplied in assembled kit form. They are disassembled to install and wire the sockets that will connect a UUT to test system driver/sensor pins, power sources, auxiliary connections, and any special customer circuits that might be required. Detailed procedures for disassembly and re-assembly of device adaptors are contained in Section 2 of the System Operating Instructions supplied with each test system.

2.2.2 Probe/Clip Preparation

2.2.2.1 GenRad Pulse-Catching Probe

The GenRad Pulse-Catching logic probe is an integrated hardware/software package that detects bad logic levels, illegal or missing pulses, pulse transitions, and oscillations. The hardware portion consists of a hand-held logic probe, connecting cable, and probe control board. The hand-held probe connects to a 14-pin LOGIC PROBE Jack. If the Pulse-Catching probe is to be used as the smart probe must be connected to the jack when the PROBE sub-mode of DIAGNOSE is run (refer to Para. 6.2.2.1).

2.2.2.2 X-1 Passive Probe

The X-1 Passive Probe may be either a Tektronix(1) Type P6101 probe or a Kurz-Kasch(2) Type LP-520-020 probe. The single P6101 connector (or the Kurz-Kasch connector marked VERIFY) is connected to one of the AUXiliary BNC jacks on the test system support shelf when the PROBE sub-mode of DIAGNOSE is run. The Kurz-Kasch connector marked POWER must be attached to the SYSTEM + 5 BNC jack, if used. The AUXiliary BNC jack selected must be wired to a dedicated Sensor pin. For example, the AUX1 connection (J23-17) can be wired through the device adaptor to driver/sensor pin 1 (J21-1). Note that the pin selected for use as a probe pin cannot be used for logic device testing and must be dedicated to the probe. In addition, the following rules apply:

1. The pin selected must be coded into the network description (refer to Para. 2.3.4.4).

2. The D/S pin selected as a probe pin must be included in the basic test program's ASSIGN LGC Output list; otherwise, the probe cannot be used in the MANUAL Sub-Mode of Diagnose. If the pin is included, the status of probed points will be listed on the MANUAL Sub-Mode digital sensor display under the probe D/S pin number. In addition, the probe pin selected must be ignored during the entire testing sequence through use of an IGNORE LGC statement.

2.2.2.3 IC Clip

The IC Clip may be used in place of the X-1 passive probe. The clip records logic states only and does not provide any pulse-catching capability. If the IC Clip is to be used as the smart clip, the clip must be connected to the optional DRM CLIP B when the PROBE sub-mode of DIAGNOSE is run (refer to Para. 6.2.2.1).

2.3 CODING THE NETWORK DESCRIPTION

2.3.1 General

The network description provides CAPS or APG with a complete description of the logic circuit to be simulated and tested. It is prepared in a standard format using a set of rules that will allow any circuit to be encoded rapidly and efficiently. Network descriptions can be prepared "off-line" using EDIT. The TEC Terminal TAB key automatically establishes the three statement field boundaries required (NAME, TYPE, and CONNECTIONS). Once a network description file has been created, it is then checked for syntax and other errors by the CAPS pre-processor described in Section 5. The pre-processor also makes the circuit connections that have been coded into the network description as the file is processed. The following paragraphs define some of the basic rules applicable to each statement field, provide a brief description of how the pre-processor connects and checks a network description, and list a set of coding guidelines that may be followed to code any network description. In addition, methods of coding a sample circuit, methods of coding the CAPS/APG special purpose library elements, and methods of coding additional information (such as special messages) through use of "percent" processing are also included.

---

(1) Tektronix, Inc., Beaverton, Oregon, 97005.
(2) Kurz-Kasch, Inc., Electronics Division, Dayton, Ohio, 45401.

2.3.2 Statement Field Rules

Network descriptions provide a method of tracing through a circuit schematic and compiling the data needed by CAPS or APG (Inputs, Outputs, IC Identifications, Interconnections, etc.) into three basic statement fields (NAME, TYPE, and CONNECTIONS). This paragraph defines the basic terms applicable to all three statement fields and lists the few rules that must be followed when entering data into any field. A discussion of how the CAPS pre-processor makes circuit connections and checks connections is presented in Para. 2.3.3, while Para. 2.3.4 contains some additional coding guidelines that amplify or expand upon the methods presented in this section.

2.3.2.1 Basic Terms Used

The following terms will be encountered as the basic rules for each of the three statement fields are presented:

1. EXTernal Signal Paths. Signal paths that connect to pins that are external to the circuit board being coded. For example, circuit paths that connect to the board edge connector pins or other circuit Input and Output paths are considered to be external signal paths. Board (or IC) Vcc and GND connections are not considered to be external signal paths and are not coded into network descriptions.

2. EXTernal Name. An alpha-numeric label that must be assigned to each external signal path being coded into a network description. Each external signal path must be assigned a different name.

3. Closed Loop. A circuit node that is to be coded into a network description and contains all fan-out paths that connect to the point being coded.

4. Intermediate Name. An alpha-numeric label that may be used, when required, to merge two disjoint closed loops together into a single closed loop during coding. Intermediate names may be used in situations where large circuit fan-out nodes extend over multiple pages of a schematic diagram so that each page of the schematic can be coded separately. For example, board set or re-set lines that extend beyond the schematic page being coded can be coded into separate disjoint closed loops, and then merged together as a single large loop by assigning the same intermediate name to each loop as it is coded.

5. Double-Cross Checks. If the point being coded into a network description has been listed previously as part of a closed loop circuit node, the pre-processor will enter a "check" mode and check that the point is connected to the next points that have been listed. It is thus possible to double-cross check the original circuit node coding by repeating the coding the next time a point on the loop is encountered.

2.3.2.2 NAME Field

The NAME Field contains the names that have been assigned to each external circuit path, IC, or special device to be coded into a network description. In the following example,

NAME	TYPE
49	EXTOUT
50	EXTOUT
IC1	7495

edge connector board pin numbers 49 and 50 have been assigned as names for two external signal paths, while the schematic diagram reference designator IC1 has been used to identify a 7495 4-bit right shift, left shift register. The following rules apply specifically to the NAME Field:

1. All external signal paths, ICs, or special devices, must be assigned names and each name must be different. Buss power connections (board or IC V_{cc} and GND connections) are not coded into a circuit and are therefore not named.

2. An alpha-numeric name can be assigned as a NAME so long as it does not exceed eight* characters or conflict with a system keyword (e.g. EXTIN). For example, the name QC010 could have been used in place of the number 49 as a functional name to identify the first external output path as the Qc Output from IC10. No spaces are permitted.

3. The EXTernal names listed are automatically assigned to the attached test system driver/sensor pins as symbolic names. For automatic assignment by CAPS, the names used must not exceed six alphanumeric characters, and the first character must be a letter, unless the driver-sensor number is equal to the NAME. Refer to the example for rule 9 in para 2.3.2.4.

4. Any name can be assigned to an IC. It is not necessary to identify the IC by the reference designator listed on the schematic diagram, however, each IC must be identified by a different name.

5. Elements within the same IC package (for example, a package that contains two flip-flops) must be listed under one name; that is, two flip-flops in the same package would not be listed as IC15A and IC15B but would both be identified as IC15.

6. IC names may be repeated as often as necessary (and in any order) so that elements within the same package, but located at different circuit points, can be coded.

7. When entering a network description, special keyboard characters that are not numbers or letters such as the @ character among others, cannot be used.

8. At least one space, a TAB, or Carriage Return (CR) must be used to define the end of the NAME Field. It is not necessary that all eight characters

---

* If PNAMES are used (refer to para 5.2.3.1 in this manual and to ASSIGN PNAME in the Test Programming Manual), the first 6 characters of each pin name used must be unique.

be used, but the NAME Field should start with column 1 so that line numbers, which are automatically assigned to the "Input Listing" Output prepared by CAPS, will be in an ordered format.

9. The TYPE Field may begin immediately after the end of the NAME Field.

2.3.2.3 TYPE Field

The TYPE Field identifies the type of external circuit path, IC or special device, that is being coded into the network description. The following rules apply specifically to the TYPE Field:

1. External circuit paths should be listed first, whenever possible, for readability. They can, however, appear anywhere in a listing and are described further in Para. 2.3.4.

2. The type name used to identify an IC (or any device in the CAPS/APG library) must agree completely with the name assigned to that device in the library. For example, catalog IC models in the library do not include SN designations, so that the correct type name for an SN7400 would be 7400. However, vendor types can be specified by entering an alphabetic designation as described in Para. 7.5.1. IC type names may consist of up to eight characters. Complete library listings that include the name assigned to each IC in the library, can be obtained through use of the Utility Program LIBLIST (refer to Section 7.)

3. The CAPS/APG user library (USERLB.CL) is normally checked first for the type name specified, and if the element cannot be located, the CAPS/APG update library (UPLIBX.CL) is checked next. If the element still cannot be located, the two CAPS/APG IC system libraries (CAPSXA.CL and CAPSX.CL) are then checked. If a positive (+) sign precedes the type name (+7400), the user and update libraries will not be checked. An asterisk (*) preceding the type name means that the non-recursive form of the element is to be used (refer to Section 7).

4. At least one space, a TAB, or Carriage Return (CR) must be used to define the end of the TYPE Field. If a CONNECTIONS Field is not required, a semi-colon (;) can be used to terminate the string at this point.

5. The CONNECTIONS Field may begin immediately after the end of the TYPE Field.

2.3.2.4 CONNECTIONS Field

The CONNECTIONS Field includes the circuit interconnections. It defines all IC interconnections, and circuit paths using the following set of basic rules:

1. IC pins are assumed to start with pin 1 and increase in an unbroken sequence. A comma (,) is used to separate pin connections and automatically increment the pin by one (two commas , , will increment the pin count by two). For example, the coding

NAME	TYPE	CONNECTIONS
IC1	7440	IC3.4, IC5.2, - - -,

may be read as "pin 1 of IC1 is connected to pin 4 of IC3". Next, the comma increments the pin count to pin 2 of IC1 and the next coding may be read as "pin 2 of IC1 is connected to pin 2 of IC5". Note that in each case, the period (.) stands for "pin number".

2. An = sign is used to specify out-of-sequence IC pins, or fields that do not start with pin 1. If, for example, the first IC pin to be coded had been pin 2 (instead of pin 1) and the pin had been connected to pin 4 of IC4, the following coding would apply:

NAME	TYPE	CONNECTIONS
IC1	7440	2=IC4.4, - - -,

Commas would then increment the pin count by one until another = sign was encountered.

3. A space is used as a separator between other points that fan-out from the point being coded. For example, if IC3.4 and IC4.5 and IC5.2 were all connected to IC1.1, the following coding would apply:

```
NAME     TYPE     CONNECTIONS

IC1      7440     IC3.4 IC4.5 IC5.2, - - - ,
                       ↑     ↑
                    SPACE  SPACE
```

4. List complete signal paths the first time they are encountered.

5. A semi-colon (;) terminates the connection string.

6. Whenever possible, list external connections first (ahead of the IC interconnections). As signal paths are traced, cross them off the schematic diagram with a marking pencil or felt tipped pen to avoid confusion.

7. When listing EXTernal connections, the first entry in the CONNECTIONS field identifies the test system driver/sensor pin connected to that circuit. For example, the following coding can be used to specify that D/S pin 2 is connected to EXTIN J1P2:

NAME	TYPE	CONNECTIONS
J1P2	EXTIN	2, - - -;

8. If double-cross checks are desired, repeat previous coding the next time a point that has already been added to a loop is reached. For example, assume that IC1.1 has already been listed as connecting to IC3.1 and IC4.5. When the connections for IC3 are listed, all of the points connecting to IC3.1 can be included in the list again if double-cross checks are desired.

NAME	TYPE	CONNECTIONS
IC1	7440	IC3.1 IC4.5, - - - ,
.	.	. .
IC3	7474	IC1.1 IC4.5, - - - ,

9. Comments may be inserted anywhere in a listing. All text entered after a slash character (/) will be inserted as a comment. A Carriage Return terminates comments.

The simple circuit shown in Figure 2-1 can be coded using rules 1 through 9. The circuit consists of a 7475 4-bit bistable latch and 74L04 hex inverters. Data at the D inputs is transferred to the Q Outputs only when the clock is high. When the clock goes low, information is retained at the Q Outputs. The Q̄ Outputs are not used in this example. Inverters are tied to each Q Output line and to the clock line. Pin 13 of IC1 is the clock line controlling the Q1 and Q2 Outputs while pin 4 controls the Q3 and Q4 Outputs. In this example, test system driver/sensor (D/S) pins are identified by circles and are assigned the same numbers as the UUT edge connector pins for convenience only. Any D/S pins available could have been assigned, and functional names could have been specified in place of the board edge connector pins listed. Note that CAPS does not require a rigid set of rules because each circuit to be coded may present different and unique problems due to circuit topology, etc., and CAPS must be capable of adapting to a variety of situations. The circuit shown may be coded as follows:

```
NAME        TYPE            CONNECTIONS


1           EXTIN       1,IC1.2;
2           EXTIN       2,IC1.3;
3           EXTIN       3,IC1.6;
4           EXTIN       4,IC1.7;
5           EXTIN       5,IC2.5;
7           EXTOUT      7,IC2.10;
9           EXTOUT      9,IC2.12;
12          EXTOUT      12,IC2.4;
13          EXTOUT      13,IC2.2;
IC1         7475        2=1,2,IC2.6 IC1.13,6=3,4,9=IC2.11,
                        IC2.13,15=IC2.3,13=IC2.6 IC1.4, 16=IC2.1;
IC2         74L04       IC1.16,13,IC1.15,12,5,IC1.4,10=7,
                        IC1.9,9,IC1.10;
#PROBE=14
```

Where all EXTernal connections are defined first, followed by the IC interconnections listed in order. While this method will present the most ordered formatting of CAPS data outputs for readability, it is not necessary that EXTernal connections be listed first, that D/S pins and functional names agree, or that ICs be listed in order. Other methods of coding are described in Para. 2.3.4.

The EXTernal CONNECTIONS coding may be read as:

1,	D/S pin 1 is connected to the EXTernal signal path NAME 1 (Which is an EXTernal INput). The comma (after the D/S pin number) is a separator and must be included.
IC1.2;	The EXTernal signal path NAME 1 is connected to IC1 pin 2. The semi-colon (;) terminates the list of connections attached to this signal path, and must be included.

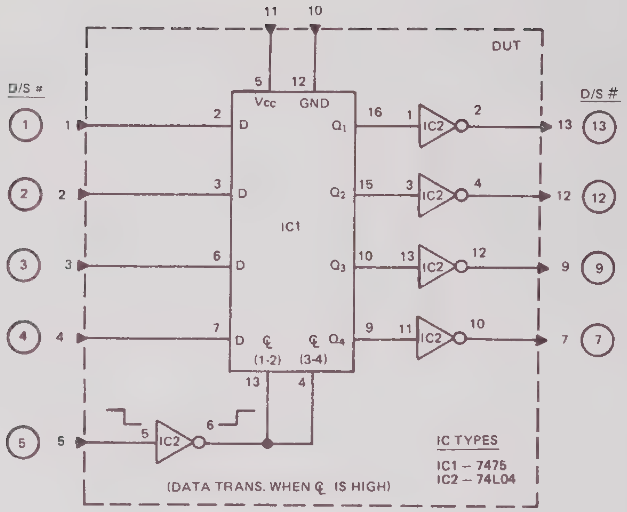

11
10
DUT
5
12
Vcc
GND
D/S #
D/S #
1
2
3
4
5
13
12
9
7
D
Q1
Q2
Q3
Q4
IC1
IC2
16
15
10
9
2
3
6
7
4
11
13
(1-2)
(3-4)
6
IC TYPES
IC1 – 7475
IC2 – 74L04
(DATA TRANS. WHEN C IS HIGH)

Figure 2-1. Coding a Simple Circuit.

The 7475 IC1 CONNECTIONS coding may be read as:

2=1,	Pin 2 is the first IC pin used and is connected to NAME 1. The comma automatically increments the pin count by one.
2,	Pin 3 is connected to NAME 2.
IC2.6 IC1.13,	Pin 4 is connected to IC2 pin 6 and (space) pin 13 of IC1. Always list complete interconnected signal paths the first time they are encountered, and separate each interconnected path with at least one space.
6=3,	Pin 6 is the next pin used and is out-of-sequence. Pin 6 is connected to NAME 3.
4,	Pin 7 is connected to NAME 4.
9=IC2.11,	Pin 9 is the next out-of-sequence pin used and is connected to IC2 pin 11.
IC2.13,	Pin 10 is connected to IC2 pin 13.
15=IC2.3,	Pin 15 is the next out-of-sequence pin and it is connected to IC2 pin 3.
13=IC2.6 IC1.4,	Pin 13 is the next out-of-sequence pin used and is connected to IC2.6 and IC1.4. Note that these connections were listed previously during the coding of pin 4 of IC1. Repeating the coding at this point will result in double-cross checking of the closed loop to determine that IC1.13 is, in fact, connected to IC2.6 and IC1.4.
16=IC2.1;	Pin 16 is connected to IC2.1. Since this completes the connections string for IC1, a semicolon (;) is added as a terminator.

The connections string for IC2 is coded next in a similar manner. The last entry defines the D/S pin number that has been assigned as a dedicated probe pin.

Intermediate (or external) names may also be used to merge two or more disjoint closed loops together into one large single loop. It is thus possible to code all signal paths associated with one page of a schematic diagram without having to be concerned with paths that extend over to other pages. If proper coding methods are followed, the two disjoint loops will be merged into a single large loop by the Pre-processor. For example, a closed loop consisting of IC1.1, IC2.2, IC3.2 (located on one page of a schematic diagram) and another closed loop consisting of IC4.2, IC5.2, IC6.2 (located on a separate page of the schematic) can be merged together through use of the intermediate name A in the following manner:

NAME	TYPE	CONNECTIONS
IC1	7440	IC2.2 IC3.2 A, - - - ,
.	.	.
.	.	.
IC4	7474	2=IC5.2 IC6.2 A, - - - ,

CAPS will merge the loops into a single large loop that consists of IC1.1 and IC2.2 and IC3.2 and IC4.2 and IC5.2 and IC6.2 and A. Note that A is simply a single merge point in the expanded loop so that listing point A alone, as a point to be cross-checked, will result in a cross-check of point A, not the loop that A is now part of.

2.3.3 Pre-Processor Operation

The following information is presented for reference purposes only and is not required reading for the new (or casual) user of CAPS. It is presented exclusively for those users who might wish a more detailed description of how the pre-processor internally "connects" and "checks" each circuit node.

Connections are always established by making complete closed loops for each circuit node specified. In the CONNECTIONS coding listed in Figure 2-2, for example, a complete circuit node has been specified from pin 2 of IC1 (the first point) to pin 1 of IC2 (the next point), and (space) EXTernal INput 6, and (space) pin 1 of IC3. The Pre-processor automatically adds the path back to pin 2 of IC1 again to complete and close the loop.

NAME	*TYPE*	*CONNECTIONS*
IC1	7474	2=IC2.1 6 IC3.1, – – –,
IC2	7440	6 IC3.1 IC1.2, – – –,
IC3	7440	IC1.2 IC2.1 6, – – –,

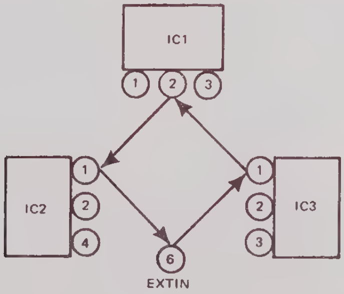


Figure 2-2. Connecting a Circuit Node.

Prior to making any connection, the Pre-processor first assembles a map of all EXTernal, intermediate, and IC names listed in the network description. Next, it examines the first line of IC CONNECTIONS coding to determine if the first point to be connected has already been connected to another loop (see Figure 2-3). If it has not, the "connection" mode is entered and a determination is made as to whether the point being connected to is already part of another closed loop. Again, if it is not part of another closed loop, the connection is finally made.

If the first point to be connected is part of another closed loop, the "check" mode is entered automatically. Connections cannot be made while the Pre-processor is in the "check" mode. This mode provides a double cross-checking feature that will determine if the first point listed is, in fact, connected to the next points specified. The number of cross-checks performed will depend upon the number of points listed in the CONNECTIONS coding. For example, in the CONNECTIONS coding for IC2 listed in Figure 2-2, the Pre-processor enters the "check" mode (instead of the "connection" mode) at the beginning because IC2.1 has already been connected to a closed loop during the coding of IC1. A check is then made to determine that IC2.1 is, in fact, connected to the points listed in the following example:

From First Point		To Next Point
IC2.1	⟶	6
IC2.1	⟶	IC3.1
IC2.1	⟶	IC1.2

Note that the first check determined that the first point had been connected to the next point specified. Another check was then made to determine that the first point had also been connected to the second point listed, etc. The degree to which a circuit node is cross-checked, therefore, will depend upon the number of additional check points that are specified each time the CONNECTIONS coding is written. For example, if the coding for IC2 had been:

```
IC2       7440       IC1.2,
```

Only the path from IC2.1 to IC1.2 would have been cross-checked. The cross-checking feature, therefore, can be as limited or extensive as a user desires for each particular board to be coded. It is recommended, however, that each closed loop be double cross-checked at least once in order to take full advantage of the pre-processor's error checking capability.

Two or more closed loops can be merged together into one single closed loop through the use of EXTernal or INTermediate names. These are alpha-numeric names that may represent edge connector pins, functional circuit signals, test system driver/sensor pins, letters that designate the end of one page of a schematic diagram and the beginning of another, etc. For example, the following coding will merge (connect) the two separate closed loops consisting of IC1, IC2, IC3 in one loop, and IC4, IC5, IC6 in another loop into a single large loop that consists of IC1 through IC6:

NAME	TYPE	CONNECTIONS
IC1	7474	2=IC2.1 A IC3.1, - - - ,
IC2	.	.
IC3	.	.
IC4	7440	A IC5.1 IC6.2, - - - ,
IC5	.	.
IC6	.	.

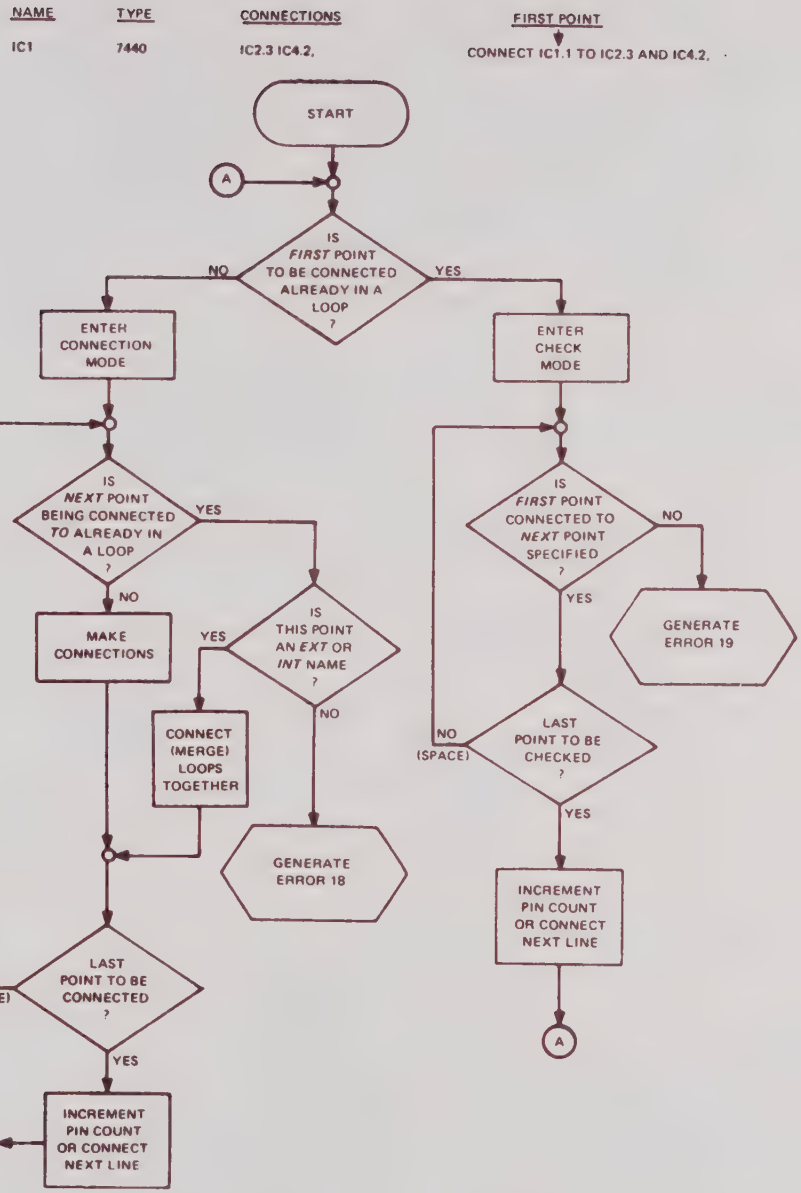

NAME
TYPE
CONNECTIONS
FIRST POINT
IC1
7440
IC2.3 IC4.2.
CONNECT IC1.1 TO IC2.3 AND IC4.2.
START
A
IS FIRST POINT TO BE CONNECTED ALREADY IN A LOOP ?
NO
YES
ENTER CONNECTION MODE
ENTER CHECK MODE
IS NEXT POINT BEING CONNECTED TO ALREADY IN A LOOP ?
YES
NO
MAKE CONNECTIONS
IS THIS POINT AN EXT OR INT NAME ?
YES
NO
CONNECT (MERGE) LOOPS TOGETHER
GENERATE ERROR 18
LAST POINT TO BE CONNECTED ?
NO (SPACE)
YES
INCREMENT PIN COUNT OR CONNECT NEXT LINE
A
IS FIRST POINT CONNECTED TO NEXT POINT SPECIFIED ?
NO
YES
GENERATE ERROR 19
LAST POINT TO BE CHECKED ?
NO (SPACE)
YES
INCREMENT PIN COUNT OR CONNECT NEXT LINE
A

Figure 2-3. Pre-Processor Connection And Check Modes.

In this example, the intermediate name A, included in the CONNECTIONS string for IC1 and the CONNECTIONS string for IC4 will result in both loops being merged into a single large loop. It should be noted that the only way to merge loops is by the use of EXTernal or INTermediate names. Merging should not be used indiscriminately, however, and might be of greatest benefit at logical break points (such as the end of one page of a schematic diagram and the start of another). Remember that whenever the pre-processor is in the "connection" mode, the point being connected to is already part of a loop, and an intermediate or EXTernal name is encountered, a merge will occur. As a result, the possibility exists that, because of a coding error, two loops that should not be merged will be merged into a single loop.

The major points to remember when coding circuit connections are:

1. Complete closed loops are established for each circuit node the first time the node is listed. Complete signal paths should therefore be listed the first time they are encountered.

2. The "check" mode checks to determine that the first point listed is connected to the other points listed. As many points as possible should therefore be repeated the next time the circuit path is encountered to take full advantage of the cross-checking feature. If complete cross-checks are not desired, however, the connections string may be reduced.

3. Connections are not made while the pre-processor is in the "check" mode.

4. Intermediate names should be assigned with care to avoid the possibility of merging circuits that should not be merged because of a coding error.

Error messages 12 and 13 are generated to point out connection errors (ERROR 12) and checking errors (ERROR 13). The following examples of erroneous coding illustrate how the messages are generated.

ERROR 12 ("Connection" Error)

```
Input Listing
Line No.          NAME      TYPE      CONNECTIONS

   010            IC1       7440      2=IC3.4,
   011            IC2       7474      3=IC5.7 IC3.4,
                                               ↑
                                            ERROR (Should be IC4.3)
```

```
ERROR#LINE#COMMENTS

 12    11   IC2.3 NOT MENTIONED AT LINE 10
            WHEN IC1.2 WAS CONNECTED.
```

ERROR 13 ("Check" Error)

```
Input Listing
Line No.              NAME      TYPE      CONNECTIONS

   010                IC1       7440      2=IC3.4 IC5.7 IC2.8,
    .                  .         .               .
    .                  .         .               .
    .                  .         .               .
   016                IC5       7440      7=IC1.2 IC2.9,
                                                    ↑
                                                 ERROR (Should be
                                                 IC2.8)
```

```
ERROR#LINE#COMMENTS

 13    16   IC5.7 FIRST CONN 10,
            IC2.9 NOT MENTIONED.
```

The names GND and VCC can also be assigned as intermediate names, and may be used to connect IC Input or Output pins to V_{cc} or GND points to simulate pullup resistors and circuit ground connections. When coded into the network description, they are automatically connected to unique CAPS added GND and VCC elements that will appear as part of the cross-reference listing only. It is also possible to use the CAPS/APG Library elements PULLUP, VCC, and GND for the same purposes, however, the elements are not added automatically by CAPS and must be coded into the network description using the procedures described in Para. 2.3.7.4.

2.3.4 Coding Guidelines

Once the basic coding rules described in Para. 2.3.2 and 2.3.3 have been mastered, the following coding guidelines can be used to code any CAPS/APG circuit. The guidelines amplify many of the unique features of CAPS and also present some alternate methods of encoding circuits.

2.3.4.1 EXTernal INput/OUTput Coding

External signal paths are paths that connect to sources that are external to the circuit board being coded (primarily Input and Output paths). External paths should be listed in the TYPE field as:

EXTIN	An EXTernal INput path that connects to an Input pin only.
EXTOUT	An EXTernal OUTput path that connects to an Output pin only.
EXT	A buss connection that connects to both Input and Output pins.

The abbreviation EXT may be used to specify any EXTernal connection (INput, OUTput, or buss). CAPS determines whether the path is an external INput, external OUTput, or buss connection. If the external path is a buss and intended to be used as an INput or an OUTput only (but not both), some core can be saved by specifying the path as an EXTIN or EXTOUT connection in the TYPE field as the network description is prepared. The abbreviation EXT can then be used to specify those connections that are known to be busses.

Each external signal path is listed only once in the network description. Do not repeat external coding.

If external signal paths that are tied together are all identified as EXT, the last path will remain as a buss (EXT) while the other paths will be listed as EXTOUT paths. This automatic assignment by CAPS can be altered by listing one of the external paths that are tied together as EXTIN. All of the remaining paths will then automatically be identified as EXTOUT.

2.3.4.2 EXTernal D/S Pin Coding

Test system Driver/Sensor (D/S) pins can be assigned in any order by listing them as the first entry in the CONNECTIONS field when coding EXTernal connections.

NAME	TYPE	CONNECTIONS
RESET	EXTIN	1, - - - - ;
CLOCK	EXTIN	5, - - - - ;

This method is preferred as it allows out-of-sequence listing of D/S pins, and also makes rapid changes to the network description possible. If it is known, however, that D/S pins are to be assigned in an unbroken order (starting with pin 1) and that the pins will not be changed, the following coding scheme can be used:

NAME	TYPE	CONNECTIONS
RESET	EXTIN	, - - - - ;
CLOCK	EXTIN	, - - - - ;

CAPS will automatically assign D/S pin 1 to the circuit path RESET and D/S pin 2 to the circuit path CLOCK, etc. If the sequence is broken (or does not start with pin 1) all D/S pins must be listed.

2.3.4.3 Basic Coding Restrictions

The maximum number of wired-AND connections in any network description is limited to 256.

2.3.4.4 CAPS X-1 Probe Sensor Pin Coding

Once a dedicated driver/sensor pin has been selected for use by the CAPS X-1 Probe (refer to Para. 2.2.2), the following entry must be included in the network description:

#PROBE=n;

Where n is the driver/sensor pin number that has been dedicated to the X-1 probe. The following rules apply:

1. Spaces cannot be inserted between the characters.
2. If the #PROBE=n;statement is not included in the .SR file, a warning message will be issued.

2.3.5 Using The Percent Processing Feature

A series of percent processing commands may be added at the end of the network description (.SR) file. Percent processing allows a user to specify optional information that can be included in listings or messages, specify IC pins and/or single nodes that are not to be faulted, etc. Table 2-1 defines the set of percent processing commands that may be used. Methods of coding the commands into the .SR file are described in the following paragraphs.

Each percent processing command string uses the following basic format:

% Percent processing commands are to follow.

WHO = WHAT;

Where WHO is a list of one or more names of externals, ICs, IC pins, and single nodes separated by commas. For example,

IC2	Affects every pin of IC2.
IC2.1	Affects IC2.1.
EXT3	Affects EXT3.
NODE IC3.1	Affects the whole node at IC3.1.
IC3,IC2.1	Affects every pin of IC3 and IC2.1.

Commas must separate each name and an = sign must separate WHO from WHAT.

WHAT is the list of percent processing commands described in Table 2-1 that will cause the actions indicated to occur. The command string must be terminated by a semi-colon (;) in each case. Table 2-1 also defines those name types (externals, ICs, IC pins, single nodes) that may be specified with each of the percent processing commands. Note that all commands do not apply to all name types. Certain commands apply to complete IC's while other commands refer to external pins, complete IC's, IC pins, and single nodes.

It may be desirable to update certain percent processing information in the .SR file when operating in the DIAGNOSE Mode. Percent Processing updates can be accomplished by running the Z sub-mode of PREPARE. The percent processing commands that are used for this updating procedure are:

NOPROBE/PROBE	(Add/Remove NOPROBE command)
NOPRINT/PRINT	(Add/Remove NOPRINT command)
NOCLIP/CLIP	(Add/Remove NOCLIP command)
NONAIL	(Insert NONAIL command)
THRESH	(Set THRESHold Levels command)
POWER	(Define POWER Pins command)
LOCATION	(Prints LOCATION message)
MESSAGE	(Displays Probe message)
NOUV/UV	(Add/Remove NO UV Command)
NOOHM	(Add NOOHM command)
CLIPD	(Add CLIPD command)
PROBED	(Add PROBED command)

In the update procedure, attributes which can come from the Library (see Table 2-1) are updated only to the extent explicitly specified in the .SR File. Other attributes are assumed to be completely specified in the .SR File. Items not explicitly specified therefore will be set to their default state regardless of previous specifications. It is not necessary to rerun the entire PREPARE Mode to update the .SR file. It is only necessary to enter the EDIT Mode, perform the update and run the Z Sub-Mode of PREPARE.

2.3.5.1 NOCLIP Command

The NOCLIP command specifies that an IC or IC pins may not be clipped due to inaccessability, etc,

Some examples of NOCLIP commands are:

IC=NOCLIP;	Do not clip IC3
IC4,IC5=NOCLIP;	Do not clip IC4 and IC5
IC3.1=NOCLIP;	Do not clip IC3 pin 1

Table 2-1
PERCENT PROCESSING COMMANDS

			Command Used Only With:			
Command	Meaning	Called From Library	EXT Pins	Complete ICs	Nodes or IC Pins	UPDATABLE Command
CLIP*	Specifies IC's or IC pins that may now be clipped that were earlier specified under the NOCLIP percent processing command.			X	X	X
NOCLIP*	Specifies IC's or IC pins that may not be clipped due to inaccessability etc.	X		X	X	X
FAMILY	Specifies that a given node or IC pins thresholds will be associated with a particular logic family (TTL, DTL, etc.) when probing.	X		X	X	
FAKE	This IC does not exist on the real board.	X	X	X		
LOCATION= "TEXT"	Prints a short message of the form "TOP PG 3" at the location(s) specified in the cross-reference (.XR) file and undetected faults (.FP) file.			X		X
MESSAGE= "TEXT1↑ TEXT2"	Displays TEXT 1 as part of the probe/FINDS message display. TEXT2 is included as part of the diagnostic printout. Both TEXT1 and TEXT2 are optional.		X	X	X	X
PRINT*	Include the IC's listed, IC pins listed, or nodes listed in the Good Board Simulation results (PRETTY) and Pin Activation (ACTIVE) listings.			X	X	X
NOSHORT	Do not model shorts between adjacent IC pins at the point or points specified.			X	X	
NONAIL	Specifies that the pin listed is not used as a nail by the bed of nails operation.		X			X
NOFUN	Do not insert "IC NOT FUNCTIONING" fault for this IC.	X		X		
UV*	Specifies that μ-voltage measurements may be performed on the IC or IC pins noted.	X		X	X	X
NOUV*	Specifies that μ-voltage measurements cannot be performed on the associated IC or IC Pins noted.	X		X	X	X

Table 2-1 (Cont'd)

PERCENT PROCESSING COMMANDS

Command	Meaning	Called From Library	Command Used Only With: EXT Pins	Complete ICs	Nodes or IC Pins	UPDATABLE Command
NOOHM	Ignores the results of ohmic measurements on an IC pin.				X	X
CLIPD	NOCLIP commands are overriden to allow the DRM to clip an IC.			X		X
PROBED	NOPROBE commands are overriden to allow the DRM to probe an IC or IC pins.			X	X	X
NOPRINT*	Do not include the ICs listed, IC pins listed, or nodes listed in the Good Board Simulation Results (PRETTY) and Pin Activation (ACTIVE) listings.	X		X	X	X
POWER	Specifies the pins to be checked during FINDS diagnosis when a node or pin is diagnosed as stuck high or low. The check is performed after it has been verified that the node is not stuck to ground or V_{cc}. The check is a continuity check between the stuck node or pin and the specified power pins.				X	X
PROBE*	This point or points can be probed.		X	X	X	X
NOPROBE*	This point or points cannot be probed.	X	X	X	X	X
THRESH	Specifies the thresholds that may be used for a given node, IC pin, or complete IC.			X	X	X
NOFAULT	Do not model SA0 and SA1 faults at the point or points specified.	X	X	X	X	
NOFAULT0	Do not model SA0 faults at the point or points specified.	X	X	X	X	
NOFAULT1	Do not model SA1 faults at the point or points specified.	X	X	X	X	

*When writing the .SR file; CLIP, PRINT, PROBE, and UV are the default states. If the user specifies NOCLIP, NOPRINT, NOUV, or NOPROBE he must then specify CLIP, PRINT, UV or PROBE to return to the original (default) state.

2.3.5.2 CLIP Command

The CLIP command specifies that an IC or IC pins may now be clipped that were earlier specified under the NOCLIP command.

An example of a CLIP command is:

```
IC4, IC5=CLIP; Clip IC4 and IC5
IC3.1=CLIP;    Clip IC3 pin 1
```

2.3.5.3 FAMILY Command

The FAMILY command enables a user to specify that a given single node or IC pin's thresholds are associated with a particular logic family (TTL, DTL, etc.) when probing. The FAMILY listed is also included in the cross-reference (.XR) file.

Some examples of FAMILY commands might include:

```
IC21=FAMILY=TTL;

IC10,IC21=FAMILY=HTTL;
```

The following logic family types may be specified:

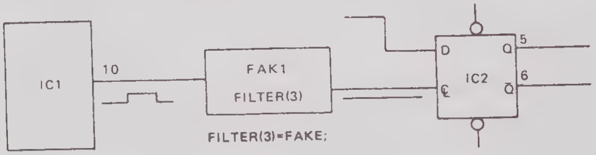
TTL, LTTL, HTTL, STTL, LSTTL, ECL, DTL, RTL, COMOS, NMOS, PMOS

The following miscellaneous logic types may be specified: MISCA-MISCE.

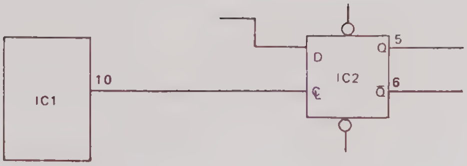
2.3.5.4 FAKE Command

The FAKE command identifies ICs that do not exist on a real board, but for some reason must be added to the network description in order to allow correct simulation of the board. For example, a FAKE filter might have to be added to the network description to eliminate a simulator pulse that does not occur on the real board or a FAKE delay element might be required to resolve a race condition.

The FAKE command is designed for use with the computer-guided probe/clip in the DIAGNOSE Mode. The probe software collapses both nodes on either side of the FAKE element to a single node in the DIAGNOSE Mode, thus eliminating the FAKE element from the actual circuit being probed. In the following example,



the CAPS simulator has generated a pulse on output pin 10 of IC1 that does not actually occur on the real board under test at the same time interval (perhaps due to circuit time delays on the real board that may not have been coded into the library model or network description properly). Unless the spurious pulse is filtered out of the simulated circuit, flip-flop IC2 will be triggered at the wrong test step. In this example, the FAKE command can be used to rapidly correct the CAPS simulator without affecting the actual board by filtering out the spurious pulse in the simulator only. Since the FILTER delay is not part of the actual board, listing it as FAKE will cause the CAPS software to effectively eliminate it during probing as in the following example:



Note that the FAKE filter has been collapsed into a single node by the probe software in the DIAGNOSE Mode and thus is not part of the actual circuit being probed.

The following rules and guidelines apply when using the FAKE command:

1. Do not FAKE Driver/Sensor pins at any time. Use transitional commands (Para. 2.5) for non-existent EXTIN pins. IGNORE non-existent EXTOUT pins or buss pins.

2. Make sure that the FAKE command does not change the nodal signature of an output pin as compared to the input pin. If the signature does change, and the probe diagnoses a failure on the collapsed node, an erroneous message of the form "BROKEN CONNECTION" will always occur. Such messages should be avoided as they serve no useful purpose, and can be confusing to an operator. If in the previous example the GenRad pulse-catching probe is being used, an erroneous "BROKEN CONNECTION" message can be avoided by creating a pulse Discrimination (.UD) file with the following entry:

```
IC1.10(ALL)MISSING;
IC1.10(ALL)ILLEGAL;
IC1.10(ALL)HALF;
```

Pulse Discrimination files are described in Section 5 and in Appendix D.

3. Do not use the FAKE command on real ICs or circuit elements that do exist on a real board under test or in a device adaptor. The BLACKBOX concept, described in Para. 2.5, is designed for use with real components that do exist on a board or in a device adaptor and are unmodelable.

4. FAKE devices are not no faulted. The NOFAULT attribute can be added, if desired.

5. The FAKE command is functionally identical to the FAKE attribute used in the GENERATE Mode.

2.3.5.5 LOCATION Message Command

The LOCATION message command can be used to output short messages that will be included in the Fault Detection (.FP) file. Messages must be brief (not more than eight characters) and may be used to inform a user of the schematic diagram page number where this element may be found, etc. The message will also be included in the cross-reference (.XR) file.

Some examples of LOCATION messages might include:

IC6=LOCATION="TOP PG 2";	IC6 is located at the top of page 2 of the schematic diagram.
IC10,IC12=LOCATION="PG 3";	These ICs are located on page 3 of the schematic diagram.

Messages must always be enclosed by quotes (") and cannot exceed eight characters (excluding quotes).

2.3.5.6 MESSAGE Command

The MESSAGE Command enables a user to specify messages that can be displayed during probing and/or messages that will be included as part of the final diagnosis printout. The format is as follows:

```
IC3.4,IC8.2=MESSAGE="PROBE RIGHT SIDE";
          OR
IC3.2=MESSAGE="↑CHECK Q1 ALSO"
```

Messages *without* the up-arrow (↑) are displayed as part of the probe message during probing. Probe messages can use up to four lines of text, however, only the *first* line will be *erased* when the next probe point is listed. Probe messages can consist of up to 71 characters on the first line, and up to 80 characters on the other three lines. If more than one line of text is used, text characters will be packed together to fill-up the maximum number of characters allowed on each line. Spaces or blank characters can be used to fill the first line if it is not desired to erase the line or to properly format lines that are to be displayed.

Messages *with* the up-arrow (↑) are displayed as part of the final diagnostic printout and are limited to 40 characters per line, however, any number of lines may be included.

Both messages can be combined into a single command string as in the following example:

$$\texttt{IC2.1=MESSAGE="}\underbrace{\texttt{PROBE RIGHT SIDE}}_{\text{Probe Msg}}\ \uparrow\ \underbrace{\texttt{CHECK Q1 ALSO}}_{\text{Diag Msg}}\texttt{";}$$

Additional up arrows (↑) in the body of the text are replaced by carriage returns when the message is displayed.

NOTE

If both pin message and an IC message are specified then the pin message takes precedence.

2.3.5.7 NOPRINT Command

The NOPRINT command is used when it is desired *not* to include ICs, IC pins, or single nodes in the Good Board Simulation Results (PRETTY) or Pin Activity (ACTIVE) listings. NOPRINT commands are also identified in the cross-reference (.XR) file. The command is ignored when using the Utility Program DISPLAY *if* the name is included in the file of elements to be output.

Some examples of NOPRINT command strings might include:

IC3=NOPRINT;	Do not list IC3.
IC2,IC5.2,NODE IC4.5=NOPRINT;	Do not list IC2, IC5.2, and the single node at IC4.5.

2.3.5.8 PRINT Command

The PRINT Command is used when it is desired to include IC's, IC pins, or single nodes in a GOOD Board Simulation Results (PRETTY) or Pin Activity (ACTIVE) listing that had been previously excluded by the NOPRINT Command.

Some examples of PRINT command listings are:

IC3=PRINT;	List IC3.
IC2, IC5.2, NODE IC4.5=PRINT;	List IC2, IC5.2, and the single node at IC4.5.

2.3.5.9 POWER Command

The POWER command specifies the power supply output pins to be checked by the Fault Identifying Nodal Diagnostic Software (FINDS) when a node or pin is diagnosed as stuck high or low. The check is performed after it has been determined that the pin or node is not stuck to ground or V_{cc}. The check is a continuity check between the stuck node or pin and the specified power pin or node.

Some examples of the POWER command are:

IC3.8=POWER;	Check continuity between stuck point and IC3.8
IC4.8,IC4.9=POWER	Check continuity between stuck point and IC4.8 and then IC4.9

2.3.5.10 NOPROBE Command

The NOPROBE command identifies those externals, complete ICs, IC pins, and single nodes that are not to be probed. Transistor bases might be examples of points that are not normally probed.

Some examples of NOPROBE commands might include:

Q5.1=NOPROBE;	Do not probe Q5.1 (the base of an NPN transistor).
IC3.4,IC2.8=NOPROBE;	Do not probe IC3.4 and IC2.8.

Points listed as NOPROBE are also identified in the cross-reference (.XR) file.

2.3.5.11 PROBE Command

The PROBE command identifies those externals, complete ICs, IC pins and single nodes, which were previously specified by the NOPROBE command, that are to be probed.

Some examples of PROBE commands might include:

Q5.1=PROBE;	The base of the NPN transistor, Q5.1, may now be probed.
IC3.4, IC2:8=PROBE;	IC3.4 and IC2.8 may now be probed.

Points listed under the PROBE command are also listed in the .XR file.

2.3.5.12 THRESHold Command

The THRESH command enables the user to specify the threshold for a given node or IC pin. The thresholds specified are included in the .XR file.

Some examples of the THRESH command are:

IC3.1=THRESH(4.7,5.2);	IC3.1 has a low threshold of 4.7, and a high threshold of 5.2.
IC3.4=THRESH(0.6,0.9);	IC3.4 has a low threshold of 0.6 and a high threshold of 0.9.

The THRESH command is used to specify unique voltages at a particular node or IC pin. This information is necessary for correct probe operation.

2.3.5.13 NOFAULT, NOFAULT0, NOFAULT1, and NOSHORT Commands

The NOFAULT and NOSHORT commands can be used to specify those ICs, IC pins, and single nodes that are not to be faulted. The points listed will be reflected in the cross-reference (.XR) file.

Some examples of NOFAULT and NOSHORT command strings might include:

IC4=NOFAULT0;	Do not model SA0 faults at IC4.
NODE IC3.1=NOFAULT1;	Do not model SA1 faults at the single node connected to IC3.1
IC2=NOSHORT;	Do not model shorts between adjacent pins of IC2.
IC3.2,IC4.5=NOFAULT;	Do not model SA0 or SA1 faults at IC3.2 and IC4.5.
EXT2=NOFAULT;	Do not model SA0 or SA1 faults at the external pin labeled EXT2.

2.3.5.14 NONAIL Command

The bed of nails option in DIAGNOSIS Mode (N=) enables the system to use each D/S as a probe. This is accomplished by reading the D/S rather than asking that the operator probe an IC pin which is connected to the D/S.

This command specifies that the bed of nails operation is disabled for the EXTernal pins listed.

Some examples of NONAIL command strings are:

EXT2=NONAIL;	External pin 2 is excluded from the bed of nails operation
EXT2,EXT4=NONAIL;	External pins 2 and 4 excluded from the bed of nails operation.

2.3.5.15 NOFUN Command

This command specifies that the fault "IC NOT FUNCTIONING" cannot be modeled for this IC.

Some examples of the NOFUN command are:

```
IC3=NOFUN;

IC6,IC10=NOFUN;
```

2.3.5.16 NOUV Command

This command identifies those IC's and IC pins that are not to be measured by the μ voltmeter during FINDS operations.

Some examples of this command are:

IC6=NOUV;	Does not measure IC 6 with the μ voltmeter during FINDS operations.
IC7.4=NOUV;	Do not measure IC 7 pin 4 with the μ voltmeter during FINDS operations.

2.3.5.17 UV Command

This command specifies those IC's and IC pins (that were previously specified by the NOUV Command) and are now to be measured when performing the μ-voltage measurements during FINDS.

Some examples of the UV command might include:

IC2=UV;	Perform -voltage measurement on IC2.
IC6.1=UV;	Perform -voltage measurement on IC6.1.

2.3.5.18 NOOHM Command

The NOOHM command is used to ignore the short detection results of a FINDS diagnosis on the IC pin or mode specified. To override the NOOHM command, a user must remove the command from the Network Description (.SR) file.

An example of a NOOHM command is:

IC2.1=NOOHM;	Ignore the ohmic measurement on IC2 pin 1.

2.3.5.19 CLIPD Command

Previously designated NOCLIP commands may be overridden to allow FINDS to clip the IC specified. CLIPD applies to FINDS only, not the guided probe.

An example of a CLIPD command is:

IC4=CLIPD;	Allow use of the DRM to clip IC4.

2.3.5.20 PROBED Command

Previously designated NOPROBE commands may be overridden to allow FINDS to probe the nodes, ICs, and IC pins specified. PROBED applies to FINDS only, not the guided probe.

Some examples of PROBED commands are:

IC4.1=PROBED;	Allow use of the DRM to probe IC4 pin 1.
IC5=PROBED;	Allow use of the DRM to probe all pins on IC5.

2.3.6 Coding a Typical Circuit

2.3.6.1 Circuit Description

The sample circuit to be coded into a network description is shown in Figure 2-4. It consists of the following major components:

1. Four 74193 synchronous 4-bit up/down counters cascaded together in a ripple, borrow/carry configuration to form a single 16-bit counter. The following pin nomenclature is used on the schematic:

A, B, C, D	Data Inputs
CLR	Clear
LD	Load Direct
CU	Count Up
CD	Count Down
QA, QB, QC, QD	Data Outputs
CO	Carry Out
BO	Borrow Out

When the 74193 LD line is low, data present at the A-D Inputs is transferred to the QA - QD Outputs. If the CLR line is set High, all Outputs (QA - QD) are forced Low. A count-up occurs when the CU input is clocked low. A count-down occurs when the CD input is clocked low.

2. Four 7495 4-bit right shift, left shift registers. These registers are cascaded together to form a single 16-bit right shift register (left shift is not used in this configuration). The following pin nomenclature is used:

RC	Right Shift Clock
MC	Mode Control
LC	Left Shift Clock
SI	Serial Input
QA, QB, QC, QD	Data Outputs

When the 7495 RC input is clocked low, a right shift occurs.

3. A 7474 dual D-type edge-triggered flip-flop with Clear (C) and pre-set (S) Inputs. A Low Input to S sets Q High. A Low Input to C sets Q Low.

4. A 7404 with six inverters. Only three of the inverters are used in the circuit.

2.3.6.2 Device Adaptor Wiring

The following device adaptor interface wiring is assumed for the purposes of preparing a network description of the sample circuit:

1. Board edge connector pins (identified by circles on the schematic diagram) are connected through the device adaptor to driver/sensor pins of the same number.

2. Driver/sensor pin 71 has been assigned as the dedicated CAPS X-1 Probe pin.

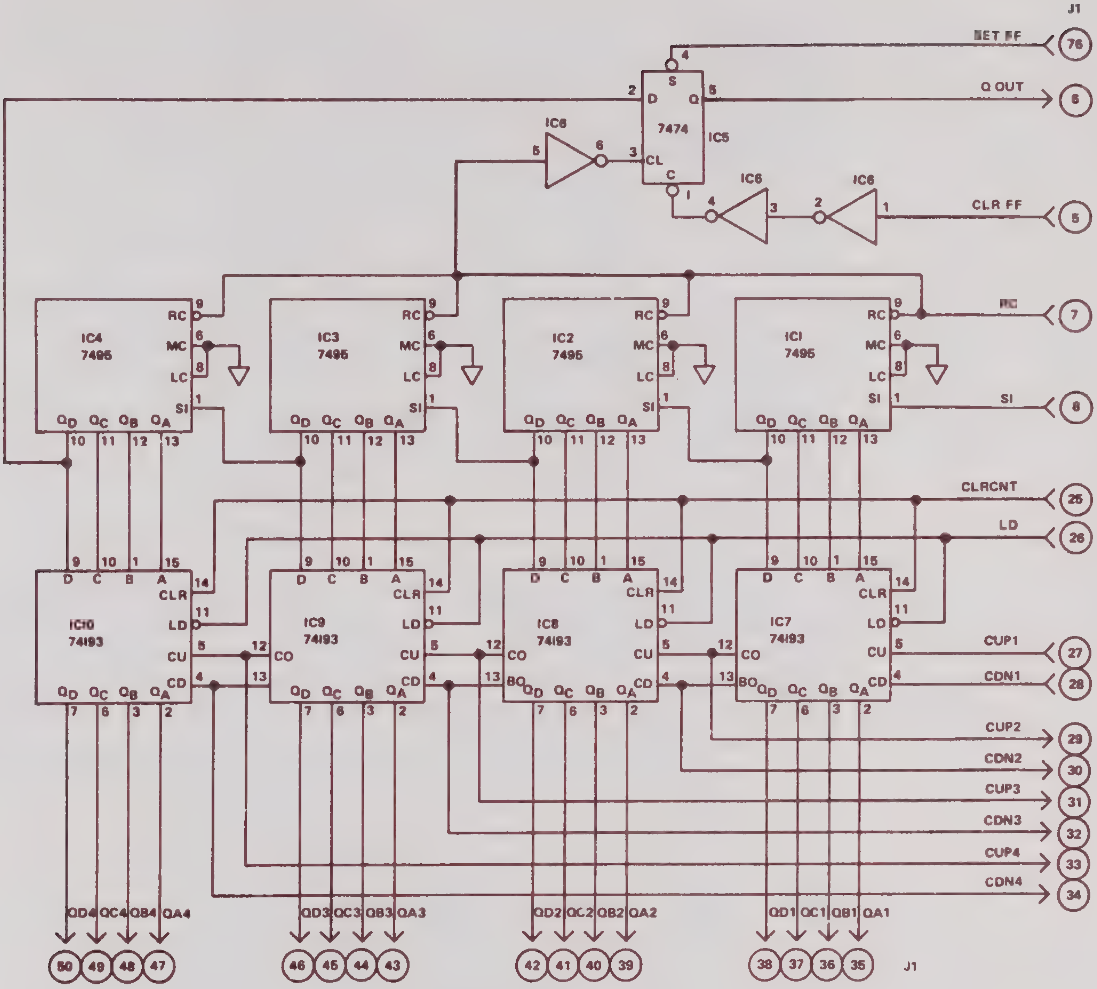

J1
SET FF
76
Q OUT
6
IC5
7474
IC6
CLR FF
5
IC4
7495
IC3
7495
IC2
7495
IC1
7495
RC
7
SI
8
CLRCNT
25
LD
26
IC10
74193
IC9
74193
IC8
74193
IC7
74193
CUP1
27
CDN1
28
CUP2
29
CDN2
30
CUP3
31
CDN3
32
CUP4
33
CDN4
34
QD4
QC4
QB4
QA4
50
49
48
47
QD3
QC3
QB3
QA3
46
45
44
43
QD2
QC2
QB2
QA2
42
41
40
39
QD1
QC1
QB1
QA1
38
37
36
35
J1

Figure 2-4. The Sample Circuit.

2.3.6.3 Preparing The CAPS Coding

The coding for the sample circuit (Figure 2-4) is shown in Figure 2-5. The following procedures should be used when coding any network description:

1. List all EXTernal connections first. As signal paths are traced, cross them off with a marking pencil or felt tipped pen to avoid confusion. Be careful and methodical when following signal paths. Use double-cross checking whenever possible.

2. Next, code each circuit IC into the network description. In the sample circuit shown, all ICs were listed in order starting with IC1. They could have been listed in any order, however.

3. After all coding has been completed, add the CAPS X-1 probe sensor number next (pin 71).

4. Add any "percent" processing features desired at this time.

```
NAME     TYPE                         CONNECTIONS
J1PIN76 EXTIN   76,IC5.4 SETFF;
J1PIN6  EXTOUT  6,IC5.5 QOUT;
J1PIN5  EXTIN   5,IC6.1 CLRFF;
J1PIN7  EXTIN   7,IC1.9 IC6.5 IC2.9 IC3.9 IC4.9 RC;
J1PIN8  EXTIN   8,IC1.1 SI;
J1PIN25 EXTIN   25,IC7.14 IC8.14 IC9.14 IC10.14 CLRCNT;
J1PIN26 EXTIN   26,IC7.11 IC8.11 IC9.11 IC10.11 LD;
J1PIN27 EXTIN   27,IC7.5 CUP1;
J1PIN28 EXTIN   28,IC7.4 CDN1;
J1PIN29 EXT     29,IC7.12 IC8.5 CUP2;
J1PIN30 EXT     30,IC7.13 IC8.4 CDN2;
J1PIN31 EXT     31,IC8.12 IC9.5 CUP3;
J1PIN32 EXT     32,IC8.13 IC9.4 CDN3;
J1PIN33 EXT     33,IC9.12 IC10.5 CUP4;
J1PIN34 EXT     34,IC9.13 IC10.4 CDN4;
J1PIN35 EXTOUT  35,IC7.2 QA1;
J1PIN36 EXTOUT  36,IC7.3 QB1;
J1PIN37 EXTOUT  37,IC7.6 QC1;
J1PIN38 EXTOUT  38,IC7.7 QD1;
J1PIN39 EXTOUT  39,IC8.2 QA2;
J1PIN40 EXTOUT  40,IC8.3 QB2;
J1PIN41 EXTOUT  41,IC8.6 QC2;
J1PIN42 EXTOUT  42,IC8.7 QD2;
J1PIN43 EXTOUT  43,IC9.2 QA3;
J1PIN44 EXTOUT  44,IC9.3 QB3;
J1PIN45 EXTOUT  45,IC9.6 QC3;
J1PIN46 EXTOUT  46,IC9.7 QD3;
J1PIN47 EXTOUT  47,IC10.2 QA4;
J1PIN48 EXTOUT  48,IC10.3 QB4;
J1PIN49 EXTOUT  49,IC10.6 QC4;
J1PIN50 EXTOUT  50,IC10.7 QD4;
IC1     7495    J1PIN8,6=GND,8=GND,J1PIN7,IC2.1 IC7.9,IC7.10,
                IC7.1,IC7.15;
IC2     7495    IC1.10,6=GND,8=GND,J1PIN7,IC3.1 IC8.9,IC8.10,
                        IC8.1,IC8.15;
IC3     7495    IC2.10,6=GND,8=GND,J1PIN7,IC4.1 IC9.9,IC9.10,
                IC9.1,IC9.15;
IC4     7495    IC3.10,6=GND,8=GND,J1PIN7,IC5.2 IC10.9,IC10.10,
                IC10.1,IC10.15;
IC5     7474    IC6.4,IC4.10,IC6.6,J1PIN76,J1PIN6;
IC6     7404    J1PIN5,IC6.3,IC6.2,IC5.1,J1PIN7,IC5.3;
IC7     74193   IC1.12,J1PIN35,J1PIN36,J1PIN28,J1PIN27,J1PIN37,
                J1PIN38,9=IC1.10,IC1.11,J1PIN26,J1PIN29,J1PIN30,
                J1PIN25,IC1.13;
IC8     74193   IC2.12,J1PIN39,J1PIN40,J1PIN30,J1PIN29,J1PIN41,
                J1PIN42,9=IC2.10,IC2.11,J1PIN26,J1PIN31,J1PIN32,
                J1PIN25,IC2.13;
IC9     74193   IC3.12,J1PIN43,J1PIN44,J1PIN32,J1PIN31,J1PIN45,
                J1PIN46,9=IC3.10,IC3.11,J1PIN26,J1PIN33,J1PIN34,
                J1PIN25,IC3.13;
IC10    74193   IC4.12,J1PIN47,J1PIN48,J1PIN34,J1PIN33,J1PIN49,
                J1PIN50,9=IC4.10,IC4.11,J1PIN26,14=J1PIN25,
                        IC4.13;
#PROBE=71;
        $
        IC10.9=NOUV;
```

Figure 2-5. A Completed Network Description For the Sample Circuit

5. Re-check the network description for obvious errors. The pre-processor will detect syntax and other errors, however, time can be saved by manually checking the coding used at this time. The most common types of errors might include:

a. Syntax errors such as omissions or extra insertions of spaces, commas, semi-colons, periods, Øs instead of Os, Is instead of 1s, etc.

b. Cross-checking errors caused by not <u>completely</u> specifying a signal path when it is <u>first</u> listed.

c. Listing ICs that are not in one of the CAPS/APG libraries. Only ICs that are listed in a CAPS/APG library can be included in network descriptions (refer to Section 7).

6. Once the network description has been completed, it may be stored as a disk file (.SR file) using the procedures contained in Section 5.

2.3.7 Special Coding Methods

The CAPS/APG library includes a series of special purpose elements that can be coded into network descriptions on an "as needed" basis to deal with special situations that may be encountered or simulate special faults. For example, it may be desired to simulate solder blobs between two adjacent tracks on a circuit board or not probe a particular circuit node as a circuit is coded. The following paragraphs describe some of the more unusual elements in the CAPS/APG library and show how they may be coded into a network description.

2.3.7.1 Coding RESET Elements

The RESET element in the library can be used to model a power-up RESET when SYSTEM POWER is applied (it generates 200 unit delays at test 0). The element can also be used to drive simulated circuits out of the "X" or unknown state in those few situations where the actual board under test is brought to a known state but the CAPS simulator remains in an "X" state due to the type of homing sequence applied to the board. The element is a single pin device that may be coded into a network description in the following manner (see Figure 2-6):

```
Z1     RESET     IC10.6;
```

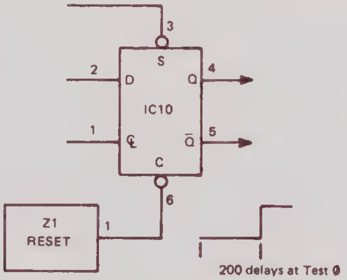


Figure 2-6. Coding a RESET Element.

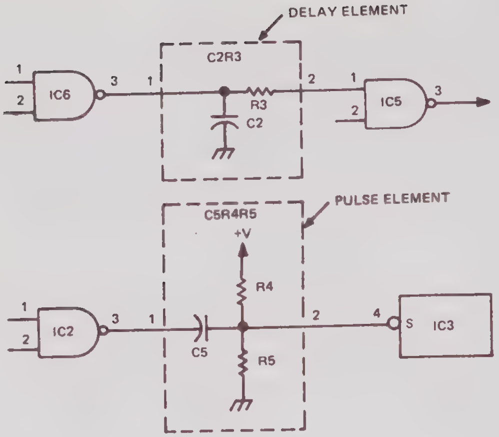
2.3.7.2 Coding Delay and Pulse Elements

A series of Delay (DEL2) and Pulse Elements (HLPN) are provided in the library for use when coding network descriptions, and can be used to model the operation of several analog components.

Delay elements are single Input and single Output devices that allow the Input to be shifted by the number of delays specified in the name. For example, a DEL2 element shifts the input by two unit delays. Delay elements can be cascaded together to form longer delays. For example, two DEL5 elements can be connected in series to shift the original input by ten unit delays. Up to a maximum of 127 unit delays can be obtained in this manner.* The maximum limit should not be exceeded, however, as "X" or unknown output states may be generated.

---

* 127 unit delays are also the maximum number of delays allowed in any one circuit before "X" or unknown output states are generated, except when long 1-shot elements are firing. With long 1-shots, the delay defaults to 4095 units. A special software patch is available on request through GenRad Software Services for those users who require an additional circuit delay capability.



```
C2R3      DEL2      IC6.3,IC5.1;

C5R4R5    HLPN      IC2.3,IC3.4;
```

Figure 2-7. Coding Pulse and Delay Elements.

Pulse elements are single Input and single Output devices that can be identified by their assigned names. For example, a HLPN provides a Positive to Negative going Output pulse when triggered by a High to Low transition. Delay and Pulse elements can be coded in the following manner (see Figure 2-7):

The name assigned to each Delay and Pulse element should always reflect actual circuit component identifications whenever possible. Error messages generated by CAPS identify failed points by the name assigned in network descriptions. If circuit component names are used, they can be easily located on the actual board under test and on schematic diagrams.

2.3.7.3 Coding One-Shot Elements

NOTE

> Do not specify one-shot time delays in the .SR file that is used to generate library elements. One-shot time delays may be specified when the board SR file that includes the library element is coded using the following procedure.

Time delays for the retriggerable one-shot (ONESHT) and for other library one-shots (such as the 9602) may be specified as follows:

1. Time delays for the ONESHT are specified in terms of typical gate delays (10ns for TTL).

2. The format used to specify one-shots in the network description is:

```
IC10     ONESHT(12)ƀ . . . . connections . . . . ;
```

where the left parenthesis occurs immediately after the IC type (ONESHT). Spaces cannot be inserted between the IC type and left parenthesis. At least one space (or TAB) must be inserted between the right parenthesis and the list connections (the ƀ indicates that a space is required). The semi-colon terminates the line. In the example listed, the one-shot will have a delay of 12. Timing for any one-shot can be specified in the same manner. For example, the coding for the 9602 in the CAPS Library might be:

```
IC5      9602(12,15)ƀ . . . . connections . . . . ;
```

Where the first one-shot in the 9602 will have a delay of 12 while the second one-shot will have a delay of 15. The lowest numbered output pin of any multiple one-shot in the CAPS IC Library is always considered to be the first one-shot in the package.

For multiple one-shots that have been inserted into the CAPS User Library (USERLB.CL), the first delay time specified will be assigned to the one-shot whose name is alpha-numerically first in the IC's SR file. For example, a one-shot named "A" would be assigned the first delay (ahead of a one-shot named "B" in the same file) even though the lowest numbered output pin has been assigned to "B".

Delays for unused one-shots in the package must also be listed. Since the one-shot will not be used, it can be assigned the minimum delay possible. For example, the coding string

```
IC5      9602(12,1)b̸ . . . . connections . . . . ;
```

will assign a minimum delay of 1 to the second unused one-shot. These delays must be specified the first time that the IC is listed. If the IC is listed again, the delay specifications cannot be included. Time delays that cross test step boundaries may also be specified by entering the letter T. For example, the coding string

```
IC5      9602(12T,15T)b̸ .... connections .... ;
```

will assign a delay of twelve Test steps to the first one-shot and a delay of 15 Test steps to the second one-shot.

2.3.7.4 Coding PULLUP, VCC, and GND Elements

PULLUP, VCC, and GND elements are all single pin library devices that can be used to simulate circuit board pull-up resistors, and circuit +V and GND points. When coding a network, IC +V and GND pins (IC power pins) are considered to be not connected, however, it may be necessary to tie an IC Input or Output pin to either +V or GND, or a pull-up resistor. The following methods of coding may be used (see Figure 2-8):

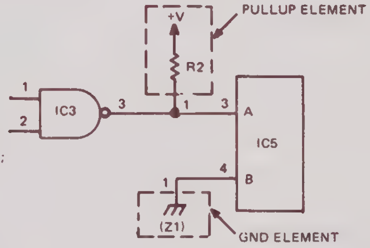


Figure 2-8. Coding PULLUP, VCC, and GND Elements.

```
R2      PULLUP      IC3.3 IC5.3;
Z1      GND         IC5.4;
```

2.3.7.5 Coding BLOB Elements

The BLOB is a two pin device that can be used to simulate a solder BLOB between two adjacent tracks on a circuit board. If the BLOB is coded into a network description, and the fault modeling B option is specified, it will be simulated as CAPS is run in the PREPARE Mode. The following coding methods may be used (see Figure 2-9):

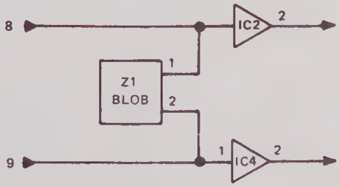


Figure 2-9. Coding BLOB Elements.

```
Z1      BLOB      IC2.1 8,IC4.1 9;
```

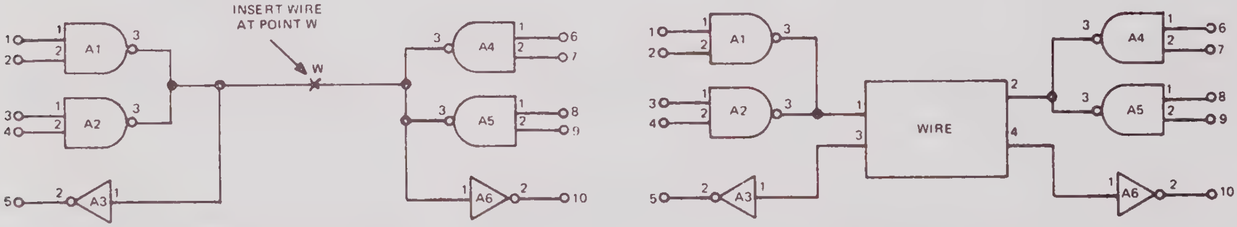
2.3.7.6 Coding WIRE Elements

Plated-through holes or weak points in wires may be simulated through use of the library element WIRE. The WIRE element can be inserted at any point on a circuit board (wherever a break is likely to occur) by including the element in the network description. When the network description is processed by CAPS in the PREPARE Mode, and the fault modeling B option is specified, a break (fault) will be modeled at that point. If an actual break occurs during logic device testing in the DIAGNOSE Mode, a

WIRE BROKEN AT XYZ

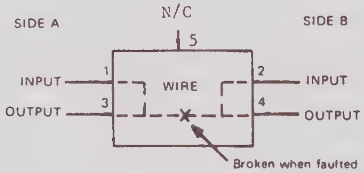
message will occur where XYZ is the name assigned to the WIRE element in the network description. It is therefore highly recommended that the name used provide some suggestion as to the 1coation of the break on the board.

The pin configuration of the WIRE element is shown in Figure 2-10. The figure also shows the element connected to a sample circuit. Figure 2-11 lists the coding necessary to insert WIRE into the sample circuit illustrated. While all Inputs and Outputs of WIRE were used in the example shown, Input and/or Output pins that are not required, need not be used. The WIRE element also has a fifth pin which is not used.



Basic Circuit Without WIRE

WIRE Inserted at Point W



WIRE Element Pin Configuration

Figure 2-10. Inserting a WIRE Element Into a Circuit.

Name	Type	Connections
1	EXT	;
2	EXT	;
3	EXT	;
4	EXT	;
5	EXT	;
6	EXT	;
7	EXT	;
8	EXT	;
9	EXT	;
10	EXT	;
A1	NAND	1,2,W.1 A2.3;
A2	NAND	3,4,A1.3;
A3	NOT	W.3,5;
A4	NAND	6,7,W.2 A5.3;
A5	NAND	8,9,A4.3;
A6	NOT	W.4,10;
W	WIRE	A1.3,A4.3,A3.1,A6.1;

Figure 2-11. Coding a WIRE Element Into a Network Description.

2.4 PREPARING A TEST PROGRAM

CAPS test programs provide the Input patterns (stimuli) necessary to exercise any logic circuit, and are prepared in the same GR High Level test language and format used for standard (non-CAPS) test systems. CAPS test programs need contain only Input patterns that will exercise internal board logic in such a manner that component failures will be propagated through the board to Output pins. Expected Output patterns are added automatically by CAPS as a simulation run is completed in the PREPARE Mode (digitally autoprogrammed) without the need for a known good physical board. The general command structure and format used to prepare test programs together with some general guidelines that can enhance program preparation are provided in the Programming Manual supplied with each test system. An example of a completed test program for the SAMPLE circuit listed in this section is shown in Figure 5-24.

2.5 CODING BLACK BOX MODELS

2.5.1 General

The "blackbox" concept allows circuits that include analog or other unmodelable components to be probed successfully. Perhaps a CAPS modeled circuit is controlled, in turn, by analog or other circuits that are difficult to model correctly. Correct modeling can be achieved by grouping the unmodelable components together in a simulated BLACKBOX that always has High Outputs. Transitional commands can then be used to change the state of the BLACKBOX outputs in the test program so that the box's correct operation can be simulated. The procedure involves entering the BLACKBOX into the CAPS/APG User Library, and then coding it into the PREPARE Mode network description (.SR file). Next, the test program must be modified to include Transitional commands (TL, TH, TX) at appropriate test steps. Each of these procedures are described in the following paragraphs.

2.5.2 Entering The BLACKBOX Into The Library

The BLACKBOX must be entered into the CAPS/APG User Library with all Output lines set to the High state so that Transitional commands will function correctly. Transitionals can pull modeled High lines low and high again, but cannot pull modeled low lines high. A typical BLACKBOX circuit that can be used to model High Outputs is shown in Figure 2-12. As shown, an OR-gate with one unused input not connected to an external pin of the BLACKBOX IC model is used to fulfill the High-State requirement. This also provides proper connectivity for the computer guided probe.

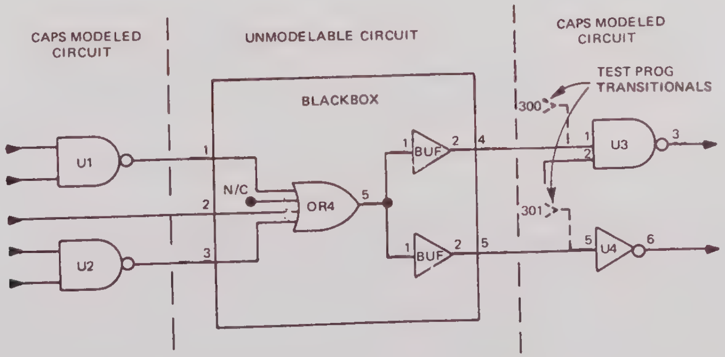


Figure 2-12. BLACKBOX Library Model.

Any of the "BOOT" elements in the CAPS/APG IC Library may be used to construct the BLACKBOX required for any application. The only requirements are that outputs within the BLACKBOX model must be buffered and must be set High. The completed model can be entered into the CAPS/APG User Library following the procedures contained in Section 7.

2.5.3 Coding The Black Box Network Description

The PREPARE Mode network description (.SR file) must list the BLACKBOX connections. For example, outputs of the BLACKBOX listed in Figure 2-12 might have been coded as follows:

NAME	TYPE	CONNECTIONS
T1	TRAN	300,BLACKBOX.4 U3.1;
T2	TRAN	301,BLACKBOX.5 U4.5;

Where TRAN is a special external connection that will be ignored by the probe and forced to be an EXTIN type. The connections column lists non-existent pins 300 and 301 as the pins that will be assigned to drive Transitionals in the test program. The BLACKBOX outputs are connected to U3, pin 1 and U4, pin 5 in the CAPS modeled circuit.

2.5.4 Test Program Transitional Commands

Once the BLACKBOX has been entered into the CAPS/APG User Library and included in the network description, test program Transitional commands can be included within CAPS recording sections of the test program. The commands may be used to drive the BLACKBOX outputs into the state specified (Low, High, or X - unknown) in order to effectively simulate BLACKBOX OPERATION. Transitionals can also be used in test-language SET PULSE commands to simulate, for example, a simple pulse generator. Transitional commands have the following general format:

```
10: IH(1,4,6);
20: IL(4);
30: IH(4) TL(300);
```

Where the BLACKBOX output line assigned to non-existent pin 300 will be driven low at test step 30.

Additional rules governing use of Transitionals include:

1. Transitional commands (TH,TL,TX) affect the CAPS simulation only, not the actual board under test.

2. Transitional pin numbers cannot be greater than 511_{10}.

3. Transitional pins cannot be included in ASSIGN LGC pin lists.

4. Actual Driver/Sensor pins may be connected to BLACKBOX Input/Output lines, if desired.

5. BLACKBOX pins cannot be faulted as a functional model does not exist. Use the % processing feature of CAPS to NOFAULT blackbox pins (refer to para. 2.3.5).

6. Never use the % processing FAKE command with BLACKBOX pins as the pins must be probed.

7. A transitional pin can be specified in a SET PULSE command (Test Programming Manual, para 3.2.6) if that pin appears in either a TH, TL or TX pin list prior to its use in the SET PULSE command. For example, the command SET PULSE-300 2; would result in 2 pulses applied to U3-1 (the transitional pin 300) shown in Figure 2-12. Note that these pulses apply only to the simulator. No driver-sensor pulses occur during board test.

2.6 DRM SENSOR THRESHOLD SELECTION

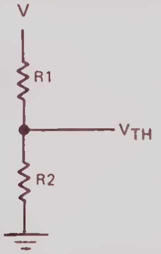
2.6.1 General

Two Diagnostic Resolution Module (DRM) options DRM I and DRM II are available. The thresholds are resistor programmed as described in the following paragraph for DRM I, and are fully programmable for DRM II (para 2.6.5).

There are up to 3 sensor threshold specifications associated with the Diagnostic Resolution Module (DRM). The specifications determine the logic threshold to be used when IC status is sensed. Eight thresholds are available to cover any logic family requirements. Four thresholds (KA-KD) are preset by GR to cover TTL, ECL, tri-state logic, as well as check that the IC clip connection to an IC is secure, with good connectivity. They must not be changed. Thresholds KE through KH are left for user requirements. Thresholds are established in the hardware by the addition of two resistors for each threshold as described below. Additionally, the voltage threshold established by the two resistors must be specified in the DRM SENSOR THRESHOLD VOLTAGES page of the SETUP mode (refer to the GR1792A-D Operating Instructions Manual).

2.6.2 DRM I Resistor Selection Formula

Two resistors form a simple voltage divider circuit as shown below. The +15V and -5V power sources are taken from the system power supplies.



$$V_{TH} = \left(\frac{R2}{R1 + R2}\right) V$$

where V= either +15V or -5V

The voltage threshold "Vth" is determined by the ratio of the two resistors (R1 and R2), multiplied by the voltage reference, +15V or -5V, whichever is wired to the "R1" side. Thus thresholds between -5V and +15V can be obtained. It is important that the sum of the resistor values selected be on the order of 20K or more to limit current flow to approximately 1 mA. Resistors that are 1/8W metal film (1%) are preferred.

Table 2-2 presents to users, a guideline of resistor values correlated to voltage thresholds between -5.00 V and +15.0 V, in 0.5 V increments. Other threshold resistor values can be interpolated in order to yield a basis for resistor selection.

2.6.3 DRM I Threshold Specification Order

For a given logic family threshold (refer to the LOGIC FAMILY THRESHOLD VOLTAGES page of SETUP mode, GR 1792A-D Operating Instructions Manual), the DRM Vth selected is found to be the one closest to the mid-point value between the logic family high and low thresholds. If two DRM Vth values are equidistant from the logic family mid-point, the first one tested is the threshold used. If no KA-KH voltage threshold is found to be within the logic family boundaries, a "NO MATCH" condition exists and the DRM IC clip is not used.

Table 2-2
DRM I VOLTAGE THRESHOLDS AND RESISTORS

```
THRESHOLD =   15. 00   R1 =      0 OHMS   R2 =  15000 OHMS
THRESHOLD =   14. 50   R1 =    500 OHMS   R2 =  14500 OHMS
THRESHOLD =   14. 00   R1 =   1000 OHMS   R2 =  14000 OHMS
THRESHOLD =   13. 50   R1 =   1500 OHMS   R2 =  13500 OHMS
THRESHOLD =   13. 00   R1 =   2000 OHMS   R2 =  13000 OHMS
THRESHOLD =   12. 50   R1 =   2500 OHMS   R2 =  12500 OHMS
THRESHOLD =   12. 00   R1 =   3000 OHMS   R2 =  12000 OHMS
THRESHOLD =   11. 50   R1 =   3500 OHMS   R2 =  11500 OHMS
THRESHOLD =   11. 00   R1 =   4000 OHMS   R2 =  11000 OHMS
THRESHOLD =   10. 50   R1 =   4500 OHMS   R2 =  10500 OHMS
THRESHOLD =   10. 00   R1 =   5000 OHMS   R2 =  10000 OHMS
THRESHOLD =    9. 50   R1 =   5500 OHMS   R2 =   9500 OHMS
THRESHOLD =    9. 00   R1 =   6000 OHMS   R2 =   9000 OHMS
THRESHOLD =    8. 50   R1 =   6500 OHMS   R2 =   8500 OHMS
THRESHOLD =    8. 00   R1 =   7000 OHMS   R2 =   8000 OHMS
THRESHOLD =    7. 50   R1 =   7500 OHMS   R2 =   7500 OHMS
THRESHOLD =    7. 00   R1 =   8000 OHMS   R2 =   7000 OHMS
THRESHOLD =    6. 50   R1 =   8500 OHMS   R2 =   6500 OHMS
THRESHOLD =    6. 00   R1 =   9000 OHMS   R2 =   6000 OHMS
THRESHOLD =    5. 50   R1 =   9500 OHMS   R2 =   5500 OHMS
THRESHOLD =    5. 00   R1 =  10000 OHMS   R2 =   5000 OHMS
THRESHOLD =    4. 50   R1 =  10500 OHMS   R2 =   4500 OHMS
THRESHOLD =    4. 00   R1 =  11000 OHMS   R2 =   4000 OHMS
THRESHOLD =    3. 50   R1 =  11500 OHMS   R2 =   3500 OHMS
THRESHOLD =    3. 00   R1 =  12000 OHMS   R2 =   3000 OHMS
THRESHOLD =    2. 50   R1 =  12500 OHMS   R2 =   2500 OHMS
THRESHOLD =    2. 00   R1 =  13000 OHMS   R2 =   2000 OHMS
THRESHOLD =    1. 50   R1 =  13500 OHMS   R2 =   1500 OHMS
THRESHOLD =    1. 00   R1 =  14000 OHMS   R2 =   1000 OHMS
THRESHOLD =    0. 50   R1 =  14500 OHMS   R2 =    500 OHMS
THRESHOLD =    0. 00   R1 =  15000 OHMS   R2 =      0 OHMS

THRESHOLD =    0. 00   R1 =  20000 OHMS   R2 =      0 OHMS
THRESHOLD =   -0. 25   R1 =  19000 OHMS   R2 =   1000 OHMS
THRESHOLD =   -0. 50   R1 =  18000 OHMS   R2 =   2000 OHMS
THRESHOLD =   -0. 75   R1 =  17000 OHMS   R2 =   3000 OHMS
THRESHOLD =   -1. 00   R1 =  16000 OHMS   R2 =   4000 OHMS
THRESHOLD =   -1. 25   R1 =  15000 OHMS   R2 =   5000 OHMS
THRESHOLD =   -1. 50   R1 =  14000 OHMS   R2 =   6000 OHMS
THRESHOLD =   -1. 75   R1 =  13000 OHMS   R2 =   7000 OHMS
THRESHOLD =   -2. 00   R1 =  12000 OHMS   R2 =   8000 OHMS
THRESHOLD =   -2. 25   R1 =  11000 OHMS   R2 =   9000 OHMS
THRESHOLD =   -2. 50   R1 =  10000 OHMS   R2 =  10000 OHMS
THRESHOLD =   -2. 75   R1 =   9000 OHMS   R2 =  11000 OHMS
THRESHOLD =   -3. 00   R1 =   8000 OHMS   R2 =  12000 OHMS
THRESHOLD =   -3. 25   R1 =   7000 OHMS   R2 =  13000 OHMS
THRESHOLD =   -3. 50   R1 =   6000 OHMS   R2 =  14000 OHMS
THRESHOLD =   -3. 75   R1 =   5000 OHMS   R2 =  15000 OHMS
THRESHOLD =   -4. 00   R1 =   4000 OHMS   R2 =  16000 OHMS
THRESHOLD =   -4. 25   R1 =   3000 OHMS   R2 =  17000 OHMS
THRESHOLD =   -4. 50   R1 =   2000 OHMS   R2 =  18000 OHMS
THRESHOLD =   -4. 75   R1 =   1000 OHMS   R2 =  19000 OHMS
THRESHOLD =   -5. 00   R1 =      0 OHMS   R2 =  20000 OHMS
```

2.6.4 DRM I Location and Access to Threshold Resistors

The resistor voltage-dividers KA-KD (thresholds 0-3) have been pre-soldered onto the 2200-4756 DRM Sensor Bd. The resistors for thresholds KE-KH (thresholds 4-7) are soldered by the user, on a component carrier which is inserted into a 24-pin socket on the same board. Perform the following steps to access and install or modify the voltage thresholds.

a. Loosen the four retaining screws securing the back cover of the system control panel, and lift the back cover off.

b. Facing the rear of the control panel, locate the 24-pin socket and component carrier for thresholds 4-7, on the 2200-4756 DRM Sensor Bd. The carrier and connector socket are mounted near the edge, approximately ¼ of the way from the right side, and marked THRESHOLD VOLTAGE. The layout of threshold resistors on the carrier is shown in J10 in Figure 2-13 below. Note that two resistors are connected for each threshold required. For threshold 4 the resistor illustrated as "R1" is connected to +15V at pin 24. "R2" is always connected to ground. This setup yields a positive threshold. To obtain a negative threshold the "R1" must be tied to -5V. This is illustrated for threshold 6.

c. The other thresholds (0-3) are mounted on the 2200-4756 DRM Sensor Bd. near TP1, in the approximate center of the board. The threshold resistors are as shown schematically in Figure 2-13.

d. Solder the resistors into the carriers being careful not to produce any solder bridges. Note that the placement for resistors onto the carrier is as follows:

Voltage Threshold	Connect "R1" To These Pins For + Thresholds	Connect "R1" To These Pins For - Thresholds	Connect "R2" To These Pins Always
4 (KE)	J10-24, 1	J10-23, 2	J10-22, 3
5 (KF)	J10-21, 4	J10-20, 5	J10-19, 6
6 (KG)	J10-18, 7	J10-17, 8	J10-16, 9
7 (KH)	J10-15, 10	J10-14, 11	J10-13, 12

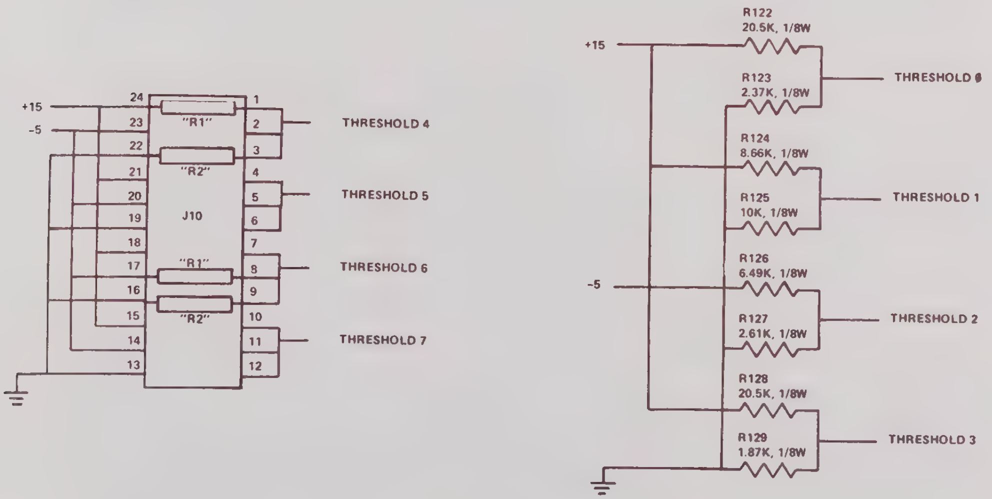

+15
-5
24
23
22
21
20
19
18
17
16
15
14
13
1
2
3
4
5
6
7
8
9
10
11
12
"R1"
"R2"
J10
"R1"
"R2"
THRESHOLD 4
THRESHOLD 5
THRESHOLD 6
THRESHOLD 7
+15
-5
R122 20.5K, 1/8W
R123 2.37K, 1/8W
THRESHOLD 0
R124 8.66K, 1/8W
R125 10K, 1/8W
THRESHOLD 1
R126 6.49K, 1/8W
R127 2.61K, 1/8W
THRESHOLD 2
R128 20.5K, 1/8W
R129 1.87K, 1/8W
THRESHOLD 3

Figure 2-13. DRM I Threshold Resistor Component Carrier Layout.

2.6.5 DRM II Sensor Thresholds

The DRM II has dual thresholds that are fully program-selectable over the range +15 V to -5 V. These thresholds are selected on the LOGIC FAMILY THRESHOLD VOLTAGES page of the SETUP mode (refer to Section 5 of the appropriate system operation manual). The software determines the logic family type of the driving IC and selects a set of thresholds accordingly.

System Operation – Section 3

3.1 INTRODUCTION

This section describes the GenRad system software and contains some basic examples of how to use it effectively. It also includes a brief description of those DEC OS/8* operating system features that are required to access GR system programs. Actual system start-up procedures are contained in Section 4.

The GenRad system software, supplied with this test system, consists of a series of major operating modes and utility programs that may be accessed from either a GenRad TEST, CAPS, APG, or HDS system monitor. The GenRad monitor, in turn, must be accessed from a DEC OS/8 Keyboard Monitor (see Figure 3-1). Both the GenRad and DEC monitors control and allow access to a unique series of modes and utility programs that can be used to rapidly prepare those files needed for digital and analog device testing, test devices, and perform other necessary file maintenance functions, when required. The major features available through each monitor include:

DEC OS/8 Keyboard Monitor

1. Immediate access to the GenRad TEST, CAPS, APG, HDS, or 1795 monitor.

2. Use of a set of DEC Concise Command Language (CCL) commands that enable a unique set of functions to be accessed quickly and easily. For example, the amount of memory actually in use by a system may be specified creation dates may be set, device DIRECTory listings may be obtained, files copied, deleted, and verified, etc.

3. Use of the keyboard monitor .R command to access a GenRad Monitor (TEST, CAPS, or APG), or to access any core image (.SV) files stored on the system device and not directly controlled by a GenRad Monitor (VPIP, PIP, ASP, etc.).

4. Use of a DEC Peripheral Interchange Program (PIP) to copy and delete files, obtain directory listings, etc.

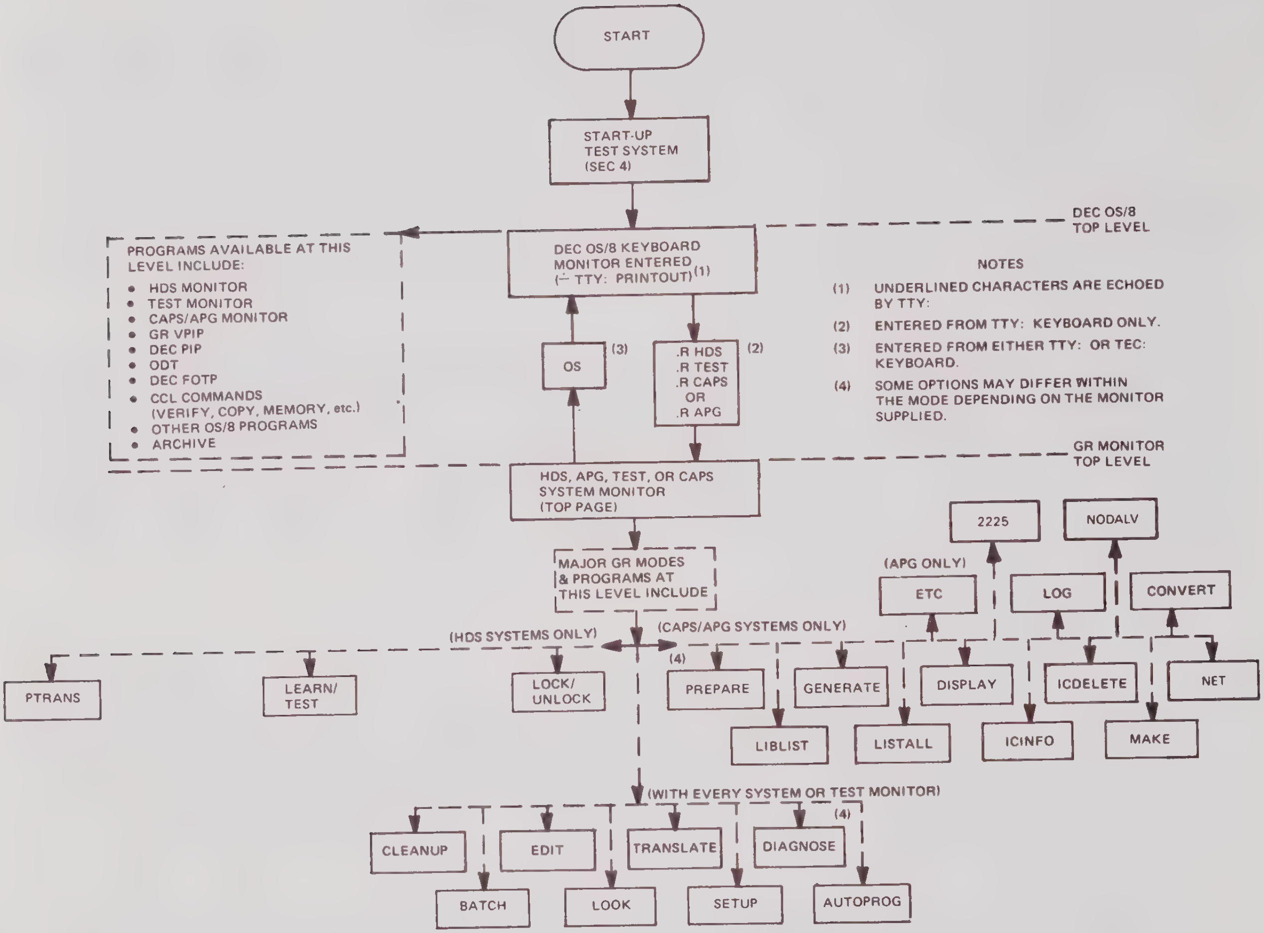
5. Use of a DEC File Oriented Transfer Program (FOTP) to transfer large files (longer than 256 blocks), perform multiple file transfers, delete files, etc. FOTP is significantly faster than PIP where transfers of similarly named files are concerned.

6. Use of a GenRad Verifying Peripheral Interchange Program (VPIP) to transfer files, verify file transfers, delete files, etc.

7. Use of a series of other optional DEC programs that can provide advanced editing features, assemble PAL8 assembly-language programs, make alterations to assembly language programs, etc. None of these programs are required for normal system operation, but are optionally available under OS/8 for use by experienced programmers, if desired.

---

* Registered Trademark of Digital Equipment Corp., Maynard, Mass.


START
START-UP TEST SYSTEM (SEC 4)
DEC OS/8 TOP LEVEL
DEC OS/8 KEYBOARD MONITOR ENTERED (← TTY: PRINTOUT)(1)
PROGRAMS AVAILABLE AT THIS LEVEL INCLUDE:
• HDS MONITOR
• TEST MONITOR
• CAPS/APG MONITOR
• GR VPIP
• DEC PIP
• ODT
• DEC FOTP
• CCL COMMANDS (VERIFY, COPY, MEMORY, etc.)
• OTHER OS/8 PROGRAMS
• ARCHIVE
OS (3)
.R HDS .R TEST .R CAPS OR .R APG (2)
GR MONITOR TOP LEVEL
HDS, APG, TEST, OR CAPS SYSTEM MONITOR (TOP PAGE)
MAJOR GR MODES & PROGRAMS AT THIS LEVEL INCLUDE
(HDS SYSTEMS ONLY)
(CAPS/APG SYSTEMS ONLY)
(APG ONLY)
PTRANS
LEARN/ TEST
LOCK/ UNLOCK
(4)
PREPARE
GENERATE
DISPLAY
ICDELETE
NET
LIBLIST
LISTALL
ICINFO
MAKE
ETC
LOG
CONVERT
2225
NODALV
(WITH EVERY SYSTEM OR TEST MONITOR)
(4)
CLEANUP
EDIT
TRANSLATE
DIAGNOSE
BATCH
LOOK
SETUP
AUTOPROG

NOTES

(1) UNDERLINED CHARACTERS ARE ECHOED BY TTY:

(2) ENTERED FROM TTY: KEYBOARD ONLY.

(3) ENTERED FROM EITHER TTY: OR TEC: KEYBOARD.

(4) SOME OPTIONS MAY DIFFER WITHIN THE MODE DEPENDING ON THE MONITOR SUPPLIED.

Figure 3-1. Overall Hard Disk System Structure.

8. Use of GenRad Archive Support Program (ASP) to extend the directory capability of an OS/8 file-structured device. Creation of archive files permits full utilization of disk file space by removing the limitation of approx. 250 directory names. Among other file storage applications, the archive facility is also used for storage of both user and system macros that can be called from within test programs.

GenRad TEST, CAPS, And APG System Monitors

All monitors provide immediate access to the following major modes and programs:

1. A GenRad Edit Program. This program allows source text in ASCII code to be created and edited (modified) with the aid of a highly efficient TEC(1) data terminal display and keyboard.

---

(1) Registered Trademark of TEC Inc., Tucson, Arizona.

2. A TRANSLATE Program. This program translates source test program files prepared by the editor in ASCII code, and written in the GenRad high-level test language, into a compact machine readable binary format. During the translation, the source text is checked for syntax and other errors. If errors occur, an operator can elect to return immediately to EDIT for any source file corrections that may be necessary.

3. A DIAGNOSE Mode. This is a large operating mode that is used to test devices and diagnose circuit failures once the proper files have been prepared. The DIAGNOSE Mode supplied with standard test systems includes a manual testing sub-mode that interacts with the system control panel to test devices in a rapid and efficient manner. The DIAGNOSE Mode supplied with the CAPS and APG Monitors includes both selectable Automatic Fault Location (AFL) and Computer Guided Probe diagnostic sub-modes in addition to the manual testing sub-mode just described. Both a GenRad pulse-catching probe and a X-1 passive probe are available for use as the computer-guided (smart) probe. In addition, a Fault Identify Nodal Diagnostic Software (FINDS) module has been added to the test system. FINDS uses a Diagnositc Resolution Module (DRM) that includes a connectivity tester and microvoltmeter that interacts with the software to resolve "stuck"-node failures to actual faults in many instances.

4. A CONVERT Program. This is a utility program that can be used to convert test programs, prepared in the older system VI GenRad 1792A/B high-level test language, into test programs that utilize the current high-level test language format. The program will also convert System VII CAPS data base files that have already been run through the CAPS N, T, and G sub-modes of PREPARE into System IX data base files without having to re-run all of the PREPARE Mode again.

5. A CLEANUP Program. This is a utility program that can be used to delete those test, listing, and temporary files that may no longer be required after completing a run.

6. A BATCH Mode. BATCH* may be used to automatically perform several preset tasks or jobs in sequence (such as the initiation of several different CAPS PREPARE Mode runs).

7. A LOOK Program. This program will Output any ASCII file to the data terminal display, line printer, or Teletype. The program may be used to look at any ASCII file without having to leave and then re-enter the TEST, CAPS, or APG Monitor in use.

8. A SETUP Program. This program sets-up the system software for the test system configuration in use (system type, type of disk controller used, number of driver/sensor cards, availability of pulse-catching probe and Diagnostic Resolution Module (DRM) options, etc.). It also establishes the threshold levels of all logic families (TTL, STTL, CMOS, etc.) to be tested with the CAPS/APG Smart Probe, and allows test system diagnostic programs to function correctly. Threshold voltages for both the DRM and Smart Probe are established in SETUP.

9. An AUTOPROGRAM Mode that automatically adds sensed logical values from a Known Good Board (KGB) to the test program.

10. A NETwork Mode that is used to specify board-test data-base files for transfer using the GenRad Network System. Refer to the 2291 Network System Operation Manual for a description of the NET Monitor Mode.

---

* The GenRad BATCH Mode is entered through a GenRad Monitor only and differs from the optional DEC Utility Program BATCH controlled by the OS/8 Keyboard Monitor.

GenRad CAPS and APG System Monitors Only.

The CAPS and APG System Monitors include a series of modes and programs in addition to the standard set of modes just described. These modes and programs consist of:

1. A PREPARE Mode. This is a large mode that is used to prepare and check all of the source inputs required by CAPS and APG (network description and test program), build a simulated model of a known good board, model selected faults, and generate the diagnostic data needed for automatic fault analysis, guided probing, and FINDS when the DIAGNOSE Mode is entered for device testing. The APG PREPARE Mode also includes options that allow the test program to be run through the Good Board Simulator in increments. Sections of the test program that work may then be automatically added to the test program.

2. A GENERATE Mode. This mode is used to generate and enter additional simulated ICs and special purpose elements into the CAPS/APG user library. Elements entered into the user library can be modeled by CAPS or APG during a simulation run.

3. A LIBLIST Program. This utility program can be used to output a list of every element in either the CAPS/APG user library or the CAPS/APG IC library. The IC library contains a series of IC models (7400, 7440, 7496, etc.) extracted from a variety of IC manufacturers handbooks, while the user library may contain unique devices inserted by the user.

4. A LISTALL Program. This utility program may be used to output all CAPS/APG listing files that have been created during a PREPARE Mode run.

5. A DISPLAY Program. This utility program Outputs the results of a CAPS/APG simulation run in the PREPARE Mode.

6. An ICINFO Program. This utility program Outputs a pin map and other useful data covering any IC or special device in either CAPS/APG library.

7. An ICDELETE Program. This utility program will delete elements from the CAPS/APG user library only.

8. A MAKE 1795 FILES Mode. This mode is used to prepare the required data base files for use by a 1795 Logic Circuit Test System. The 1795 system uses floppy diskettes or hard disks for mass storage, therefore the data base files can be transferred onto either device through use of the MAKE Mode.

9. A LOG Mode. The LOG Mode provides the facility for logging data learned during DIAGNOSE Mode execution. This includes AFL and guided probe information. The logged binary data is stored in pre-allocated files for later retrieval.

10. ETC Mode. This is an APG troubleshooting mode that enables a user to track down indeterminate "X" states, positive or negative going transitions and pulses that may have been noted in the PRETTY file output. The mode can also be used to provide a connectivity check to determine which output pins are affected by active nodes, and generate Boolean equations that can be used to determine the input states required to drive any pin or node into a particular high or low state.

11. Nodal Verification (NODALV) Mode. This mode is used to automatically prepare Pulse Discrimination (.UD files) for use with the GenRad pulse-catching probe in the DIAGNOSE Mode.

12. 2225 Mode. This mode is used to convert CAPS test programs and network descriptions into a format usuable by the 2225 Portable Service Tester.

Methods of accessing and using major modes and programs controlled by each GR Monitor are described in the following paragraphs. The detailed use of each GR Mode or program is covered in the following publications:

CAPS/APG Operation

- PREPARE
- DIAGNOSE (for CAPS and APG)
- GENERATE
- DISPLAY
- MAKE
- ETC
- 2225 MODE
- LIBLIST
- LISTALL
- ICDELETE
- ICINFO
- LOGGING
- NODALV

System Operation

- EDIT
- TRANSLATE
- AUTOPROGRAM
- DIAGNOSE (for TEST)
- CONVERT
- SETUP

COVERED IN BOTH PUBLICATIONS

- GR VPIP
- LOOK
- CLEANUP
- BATCH
- ARCHIVE

All HDS Monitor Modes (PTRANS, LEARN/TEST, LOCK/UNLOCK, etc.) are described in the System Operation Manual supplied with 1795 or 1799 test systems.

3.2 DEC OS/8 OPERATING SYSTEM

3.2.1 General

Programs and modes directly controlled by OS/8 may be accessed through the OS/8 keyboard monitor. Programs and modes directly controlled by a GenRad Monitor (TEST, CAPS, or APG) must be accessed through the GenRad Monitor involved and are described in Para. 3.3. A brief description of those features of OS/8 that may be useful in accessing system programs and modes is provided in the following paragraphs together with a description of any GenRad programs or commands that are controlled directly by the keyboard monitor (VERIFY, VPIP, and ASP). A complete description of all DEC OS/8 programs is contained in the DEC OS/8 Handbook supplied with each test system.

3.2.2 Keyboard Monitor

The keyboard monitor* is entered whenever a test system start-up occurs. It can also be entered from any system program under the direct control of the keyboard monitor, by depressing the Teletype CTRL and C keys (hold down the CTRL key while activating the C key). When the keyboard monitor is ready to accept commands, a dot (.) printout occurs. The keyboard monitor accepts commands only from the Teletype keyboard. Some of the more important functions and programs that are under the control of the keyboard monitor include:

1. The DATE command. This is a CCL command that assigns a date to all files created, for record purposes. Some modules and programs require that a date be established before the program can be entered. Dates remain in effect until another test system start-up occurs. The command has the following format:

```
.DA DAY-MONTH-YEAR
```

---

*The keyboard monitor may also be entered from a GenRad Monitor through use of the OS command.

For example, the command

```
.DA 1-APR-79
```

would establish a date of 1 APRIL, 1979.

2. The R Command. This command allows core image (Save) files located on the system device to be accessed. For example,

.R CAPS (or.CAPS)	Calls the CAPS Monitor.
.R TEST (or.TEST)	Calls the TEST Monitor.
.R HDS (or.HDS)	Calls the HDS Monitor.
.R APG (or.APG)	Calls the APG Monitor.
.R VPIP	Calls the GR VPIP Program.
.R ASP	Calls the Archive Support Program.

Commands listed in parenthesis are CCL Commands that may be executed without specifying the .R prefix, if desired.

The Teletype RETURN (↓) or ALTMODE keys may be used to activate each command. RETURN causes a Teletype carriage return and line feed to occur while the ALTMODE key does not. ALTMODE also echoes back a $ character. An ESC key may also be used in some machines and is equivalent to ALTMODE.

3. CCL Commands. A series of extremely useful CCL commands are available with this version of OS/8 and are described in detail in the DEC OS/8 Handbook. Some of the more important commands include:

CCL Command	Function
.SQ SYS:↓	SQUISH the SYS: device to remove any embedded empty files. The Teletype echoes SAME DEVICE, ARE YOU SURE? A user must answer Y (Yes) to proceed. Any other character aborts the command.
.SQ↓	SQUISH the default device (DSK:).
.DIR SYS:/A↓	List the SYS: device DIRECTory to the TEC:. Alphabetize the directory (/A).
.DIR↓	List the default device directory (DSK:) by TEC:. Do not include empty file lengths.
.DIR*.BC↓	List all files with a .BC extension on the default device (DSK:) regardless of the first part of the file name.
.DIR LPT:< SYS:/A↓	Output an alphabetized directory to the LPT:
.DEL SYS:NAME.BC↓	Delete the file NAME.BC from the SYS: device. The Teletype echoes FILE DELETED: NAME.BC

CCL Command	Function
.MEMORY↓	List the amount of MEMORY available for use in this test system. A teletype printout of the amount of memory will occur. This command may also be used to limit the amount of memory that can be used by the system software. For example, listing MEMORY n↓will limit the amount of memory in the following manner.

n	memory
0	Use all memory
1	8K only
2	12K only
3	16K only
4	20K only
5	24K only
6	28K only
7	32K only

NOTE

The CCL Command, .MEMORY, presently provided will only indicate up to 32K of memory.

If the amount of memory has been limited through use of the MEMORY n command, then the teletype printout will include both the amount of restricted memory and the total system memory available. For example,

```
20K/32K MEMORY
```

specifies that a 32K system has been restricted to 20K through use of the MEMORY 4 command.

CCL Command	Function
.HELP CCL↓	Displays the HELP CCL file on TEC:. HELP files are located on TEST/HDS disks only.
.COP SYS:< DSK2:NAMEA↓	Copy the file NAMEA from DSK2: to the SYS: device.
.REN SYS:NAMER< SYS:NAMEA↓	RENAME the SYS: file NAMEA as NAMER. Note that RENAME can only re-name files on the <u>same</u> device.
.LOOK DSK2:NAMEA↓	Displays the file NAMEA on TEC:
.LIST DSK2:NAMEA↓	Lists the file NAMEA by LPT:.
.VERI SYS:<DSK2:NAMEB↓	Verify that the file NAMEB and all files that follow on DSK2: are identical to files of the same name on SYS:.
.ZER DSK2:↓	ZERO the device directory of DSK2:.

GenRad Modifications to the CCL commands are described further in Para. 3.2.4. L commands can be disabled (or enabled) from the keyboard monitor through use of e commands:

.CCL ↓	To disable CCL commands.
.R CCL ↓	To enable CCL commands.

3.2.3 Command Decoder

The command decoder interprets (decodes) command lines, entered by an operator, to indicate Input and Output devices or files and options desired. The command decoder is entered automatically when the system program, accessed through the keyboard monitor, is ready for use as indicated by an asterisk (*) printout. The command decoder (like the keyboard monitor) accepts commands only from the Teletype keyboard.

Files may be Output to or Input from both "file structured" and non-file structured devices. "File structured" devices are direct access devices that have been formatted into 256 word blocks so that read or write operations can be accomplished from any desired block. Disk drives are examples of "file structured" devices. Non-file structured devices include all other system devices such as the Teletype, data terminal, paper tape readers, punches, and line printers. The following Output/Input devices are available for use with this test system and may be specified in command decoder command lines:

FILE STRUCTURED DEVICES

Top (or only) Disk Drive

SYS: (or DSK0:)	The System device. Holds pre-recorded system modules and programs. May also contain test files.
DSK: (or DSK1:)	The default device. This device is accessed if a device is not listed in a command line. Provides additional storage space for test files, etc.

Bottom Disk Drive (Optional)

DSK2:	Provides additional storage capacity for test files, etc.
DSK3:	Provides additional storage capacity for test files, etc.

Magnetic Tape Storage*

DTA0:	Provides additional storage capacity for test files, etc.
DTA1:	Provides additional storage capacity for test files, etc.

NOTE

Magnetic tape drives cannot be assigned as SYS: devices in this test system.

---

*Replaced by Floppy Diskette Storage in newer test systems.

Floppy Diskette Storage (Optional)

FLP0: FLP1: FLP2: FLP3:	Stores data base files for 1795 test system use. Data base files are output through use of the CAPS/ APG MAKE Mode. Also provides additional storage capacity for test files, etc.

NOTE

Floppy Diskettes cannot be assigned as SYS: devices in this test system.

Other Disk Drives (Optional)

DSK4: and DSK5:	Provides additional storage capacity for test files, etc.

If DSK4: and DSK5: are used, it will be necessary to activate device handlers for these devices by running the DEC OS/8 BUILD program. When BUILD is run, device handlers may be activated and/or deactivated through use of a series of BUILD commands (INSERT, DELETE, BOOT, etc.). Complete instructions for running BUILD and using the various BUILD commands are contained in the DEC OS/8 Handbook supplied with each test system.

Test systems are normally shipped with a complement of 15 active device handlers. In order to make room for the DSK4: and DSK5: device handlers, it will be necessary to deactivate two of the active handlers. The following commands may be issued from the OS/8 Keyboard Monitor to run BUILD, activate the DSK4: and DSK5: handlers and deactivate the FLP0: and FLP1: handlers:

```
.RU SYS:BUILD ↓
$IN RK05:DSK4-5 ↓
$DEL FLP0-1 ↓
$BO ↓
SYS BUILT response.
.SA SYS:BUILDA ↓
```

Note that the updated version of BUILD was saved under another name (BUILDA) so that the original handler configuration can be easily reactivated again by calling BUILD and issuing the BOOT command. For example,

```
.RU SYS:BUILD ↓
$BO ↓
SYS BUILT response.
```

will reactivate the FLP0: and FLP1: handlers while deactivating the DSK4: and DSK5: handlers.

NON-FILE STRUCTURED DEVICES

TTY:	Teletype
LPT:	Line Printer
TEC:	Data Terminal Display
PTP:	H.S. Paper Tape Punch
PTR:	H.S. Paper Tape Reader
CDR:	Card Reader

3.2.4 File Management Under CCL

A series of CCL commands are available to assist the user in file management (COPY, DIRECT, VERIFY, SQUISH, etc.). Each of the commands link to a DEC or GenRad system utility program to accomplish their assigned tasks and then return again to the keyboard monitor. Table 3-1 lists the file management CCL commands, the program used, and where additional information covering the program is located.

Table 3-1

FILE MANAGEMENT CCL COMMANDS

Command	Program Used	Detailed Description
DELETE	DEC FOTP	DEC OS/8 Handbook
COPY	DEC FOTP	DEC OS/8 Handbook
DIRECT	DEC DIRECT	DEC OS/8 Handbook
VERIFY	GR VPIP	Para. 3.2.5
SQUISH	GR VPIP	Para. 3.2.5
ZERO	GR VPIP	Para. 3.2.5
RENAME	DEC FOTP	DEC OS/8 Handbook

The VERIFY, SQUISH, and ZERO commands each link to the GR VPIP program described in Para. 3.2.5 to accomplish their assigned tasks before returning to the keyboard monitor. Some useful file management commands might include:

Command	Function
.VERI SYS: < DSK2:NAMEA/O ↓	Verify that the file NAMEA on DSK2: is identical to the same file on SYS:. /O specifies only this file and no others. It is not necessary to list Output device file names with VERIFY as Output device files cannot be re-named by VERIFY.
.VERI SYS: < DSK2:NAMEF ↓	Verify that the file NAMEF on DSK2: and all files that follow are identical to the same files on SYS:.
.VERI SYS: < DSK2: ↓	Verify that all files on DSK2: are identical to the same files on SYS:.
.VERI DSK1: < DSK2:NAMEL/N/O ↓	Transfer (copy) the file NAMEL from DSK2: to DSK1: and then verify that the file is identical to the same file on DSK1: (performs both a non-destructive file transfer and verify operation in a single command line. Note that it is not necessary to list Output device file names under VERIFY as files cannot be re-named.
.VERI DSK1: < DSK2: /N ↓	Transfer (copy) all files from DSK2: to DSK1: and then verify the file transfer.
,VERI DSK1: < DSK2: /S ↓	Transfer (copy) all files from DSK2: to DSK1: and then verify the file transfer. This is a destructive transfer (files on DSK1: are deleted prior to the transfer).

.SQ SYS: ↵	SQUISH the SYS: device to remove any embedded empty files. An ARE YOU SURE? message occurs prior to running SQUISH. A user must answer Y (Yes) to proceed. Any other character aborts the operation.
.SQ ↵	SQUISH the default device (DSK:) to remove any embedded empty files.
.ZERO DSK1: ↵	ZERO and DSK1: device directory. If an attempt is made to ZERO the system device, a ZERO SYSTEM? message occurs. A user must answer Y (Yes) to proceed. If the syntax is in error (for example, not specifying a colon after the device) an ILLEGAL SYNTAX message will occur.

3.2.5 GenRad VPIP Program

3.2.5.1 General

The GenRad Verifying Peripheral Interchange Program (VPIP) is a powerful file management tool that may be entered directly from the DEC keyboard monitor or called through use of the CCL VERIFY, SQUISH, and ZERO commands to perform specific tasks. The following services are offered by VPIP:

1. Perform destructive file transfers between different file-structured devices. Files are transferred from the Input device to the Output device and overlay (destroy) any Output device data that may be present. The Output device directory is then updated to include only the files that have been input. Empty files are not transferred.

2. Perform a destructive file transfer on the same file structured device to remove (squish-out) any embedded empty files that may be present. Files are transferred on a one-to-one basis as the embedded empty files are removed.

3. Perform non-destructive file transfers between devices. All (or selected) files may be transferred from an Input device to an Output device. Files are placed in either the "best fit" locations or the "largest empty" locations available during the transfer (depending on the options selected). As files are transferred, the device directory is updated to include all files present on the Output device. Input device files with different names are added to any Output device files that may be present. Files on the Input device replace files on the Output device only if the file names are identical.

4. Verify read/write operations and directory entries to check the accuracy of file transfers as they are performed.

5. Verify and identify any files on the Input device that are not identical to files on the Output device.

3.2.5.2 Program Structure

VPIP is a two-pass program that differs from PIP in that transfers can be Verified to determine that read/write operations have been performed successfully and that information in the device directory is correct after the transfer has been completed. When VPIP is in use, both destructive and non-destructive file transfers occur during pass 1 (depending on the group of options selected). In addition, one option (T) may be used to verify file transfers during pass 1 by reading inputs twice and then reading the same data back after performing each write operation. During pass 2, a verification of the device directory is performed by using file lookups to determine if the data in the lookup file(s) is identical to the data in the Input file(s). File transfers do not take place during pass 2.

3.2.5.3 Terms Used

The following definitions of frequently used terms may be of value in understanding how VPIP transfers files:

1. File Structured Device. A directory oriented device that has been formatted into 256 word blocks so that read or write operations can be accomplished from any desired block. Disk cartridges are examples of file structured devices. File structured devices have device directories that include all permanent, temporary, and empty files stored on the device.

2. Non-File Structured Device. A device that does not have a device directory. Teletypes, line printers, punches, data terminal displays, and paper tape readers are examples of non-file structured devices.

3. File. A collection of data that is treated as a single unit. The data may be in any format (ASCII, binary, core image, etc.).

4. Permanent File. A file that has been closed (assigned a fixed number of blocks on a file structured device and is included in the device directory).

5. Empty File. A contiguous sequence of blocks not occupied by permanent files. When permanent files are deleted, empty files (holes) are created. Empty files are included in device directories.

6. "Best Fit" Location. Files may be placed wherever they will fit best on a device, thus tending to fill-in empty locations (holes). Files may be scattered throughout the device after the transfer.

7. "Largest Empty" Location. Each file may be placed in the largest empty location (hole) available. Files will usually be grouped together at the end of the device after the transfer.

8. Non-Destructive File Transfer. Files on the Input device with different names are added to any files that may be present on the Output device. Files on the Input device replace files on the Output device only if the file names are identical.

9. Destructive File Transfer. Files on the Input device overwrite existing files on the Output device as the transfer occurs. The Output device directory is then updated to include only the new files that have been transferred. All files (that existed prior to the transfer) are deleted from the Output device directory.

10. Directory Order. Names assigned to permanent files are listed in the device directory together with the number of free blocks available for additional file storage. Optional data that includes the lengths of files, file creation dates, and a list of empty files may also be provided. Directory Order is the order (from top to bottom) that file names appear in the device directory. When files are transferred in "directory order", they are transferred in the order that they appear in the "Input device" directory (from top to bottom) only.

3.2.5.4 Accessing VPIP

VPIP may be accessed from the keyboard monitor by entering the following command through the Teletype keyboard:

```
.R VPIP ↓
```

VPIP command strings have the follwing general format:

```
*DEV:<DEV:NAME1,DEV:NAME2, . . . . /OPTIONS ↓
```

Where:

*DEV:<	Is the Output device (Output files to this device). The < separate Output specifications (to the left) from Input specifications (to the right). The ← symbol may also be used as a separator and is equivalent to the < symbol shown. Only one Output device can be specified for VPIP.
DEV:NAME,	Is an Input file name and device. File names are separated by commas. If a device is not specified, the last Input device listed is assumed. If no Input device is listed, the default device (DSK:) is assumed. File names can consist of up to six alpha-numeric characters optionally followed by a period and two character extension (NAME.XX). Up to nine Input files located on the same or different devices and separated by commas can be listed in a single command string.
/OPTIONS	Options are letters or numbers that perform specific functions as the command string is executed. Table 3-2 lists the options available. Options can be listed in the slash format /V/N/O or enclosed in parenthesis (VNO).
↓	Is the RETURN key used to activate a command string.

VPIP may also be entered through the use of the CCL commands VERIFY, SQUISH, and ZERO described in Para. 3.2.4. Each of the CCL commands automatically invoke the following VPIP options when they are used:

CCL Command	Operation
.VERIFY	Invokes /V.
.SQUISH	Invokes /V/S. If an Input Device is not listed, DSK: is assumed. If an Output Device is not listed, the first Input Device is used and /T/H are invoked.
.ZERO	Accepts the Device specified as the Output Device. No Input Device is allowed, and the /V/S/T/H options are invoked.

After completing a CCL operation, the OS/8 Keyboard Monitor will be entered. Other VPIP options may be combined with CCL commands to perform any of the tasks listed in Table 3-2. Some CCL command line examples are provided in Para. 3.2.4.

Table 3-2

GenRad VPIP OPTIONS

Option	Function
	CHECKING NON-IDENTICAL FILES
/V	Use the GenRad VPIP Options listed in this table. /V should be specified with each option listed. If used along, a Pass 2 Verification only will occur (device directory file lookups). File transfers do not take place. Non-identical device files are identified and listed. /V is automatically invoked by the CCL command VERIFY.
/O	Perform a Pass 2 Verification (device directory file lookups) of the file names listed only.
	DESTRUCTIVE FILE TRANSFERS
/S	Perform a destructive file transfer, moving files from the Input device to the Output device, overwriting existing files on the Output device as the transfer occurs. Empty files are not transferred. After the transfer, the Output device directory will contain only those files that were transferred. Files are transferred on a one-to-one basis in the order that they appear in the Input device directory only. If a file name is specified, all files from that point through to the end of the directory will be transferred. When /S is specified, an ARE YOU SURE? message will occur. A user must answer Y (Yes) to proceed. Any other character aborts the command.
/O	Transfer only the file names specified (or if no files are specified, transfer the first file listed in the Input device directory for each device specified). Multiple files must be listed in the order that they appear in the Input device directory when performing a /S transfer on the same device (self-squish). If this rule is not followed, verification errors will occur.
/T	Verify read/write operations during Pass 1 by reading Inputs twice and then reading the same data back after performing each write operation. Used when Pass 2 verifications cannot be performed or if it is desired to identify errors during Pass 1.
/H	Halt the Pass 2 Verification. Directory file lookups are not performed. Pass 2 Verifications cannot be performed when self-squishing devices (for example SYS:<SYS:).

Table 3-2 (Cont'd.)

Option	Function
	NON-DESTRUCTIVE FILE TRANSFERS
/N	Perform a Non-destructive file transfer, moving files from the Input device to the Output device. Files may be inserted into the "best fit" locations available if /D is also specified; otherwise, files are inserted into the "largest empty" locations available. As files are transferred, the device directory is updated to include all files present on the Output device. Input device files with different names are added to any Output device files that may be present. Files on the Input device replace files on the Output device only if the file names are Identical. Files are transferred on a one-to-one basis in the order that they appear in the Input device directory only. If a file name is specified, all files from that point through to the end of the directory will be transferred.
/O	Transfer only the file names specified (or if no files are specified, transfer the first file listed in the Input device directory for each device specified).
/D	Used with /N to Delete old files of the same name prior to the file transfer. If /D is not specified, the old file will not be deleted until after the file transfer. Files transferred with /D are placed in the "best fit" locations available. Files transferred without /D are placed in the "largest empty" locations available.
/T	Verify read/write operations during Pass 1 by reading Inputs twice and then reading the same data back after performing each write operation. Used when Pass 2 Verifications cannot be performed or if it is desired to identify errors during Pass 1.
/H	Halt the Pass 2 Verification. Directory file lookups are not performed.
	VERSION PRINTOUTS
/W	List the current version of VPIP that is being used.

3.2.5.5 Using the GenRad VPIP Options

The following examples of command lines that include the VPIP options are provided for a user's convenience. Table 3-2 lists the VPIP options available.

LIST NON-IDENTICAL FILES

`*DSK1:<DSK2:/V` ↲	Run the Pass 2 Verification only (directory lookup). Lists all files on DSK2: that do not have identical files on DSK1. File transfers do not take place.
`*DSK1:<DSK2:NAME1/V/O` ↲	Run the Pass 2 Verification (directory lookup) of only the file names listed. File transfers do not take palce. In the example listed, a check is made to determine if the entry for DSK2:NAME1 is the same as that for DSK1:NAME1.

DESTRUCTIVE FILE TRANSFERS

NOTES FOR /S

- Only the Input files being transferred will appear on the Output device directory *after* completing a /S transfer. *All* Output device files that existed prior to the transfer will be lost.

- Be careful when specifying *multiple* Input devices with /S. *All* files specified as Input (including those specified by multiple Input specifications, with or without the /O option) will appear on the Output device after the transfer, even if that means *multiple* files of the *same* name.

- Be careful when specifying *multiple* Input *files* with /S. All file transfers (including self-squishes) are accomplished sequentially by file. The first Input specification file listed is transferred first, followed by the next, etc. If Input files are to come from a device which is also an Output device (a self-squish), care must be exercised to avoid writing over a file before it can be used as an Input file. A useful (but not necessary) rule is to only specify *multiple* Input files with the /O option in the order that they appear in the device directory (directory order) when performing a /S transfer on the same device (self-squish).

`*DSK1:<DSK2:/V/S` ↓	Transfer all files from DSK2: to DSK1: overwriting any data that may be present on DSK1:. Update the DSK1: directory to include only the files being transferred, and perform a Pass 2 Verification. Does not transfer empty files.
`*DSK1:<DSK2:NAME1/V/S` ↓	Transfer files from DSK2: to DSK1: *starting* with NAME1, and perform a Pass 2 Verification. All files from NAME1 through to the end of the directory are transferred in "directory order".
`*DSK1:<DSK2:NAME1,NAME2/V/S/O` ↓	Transfer *only* the file names specified. After the transfer, DSK1: will contain *only* the files NAME1 and NAME2 as copied from DSK2:. A Pass 2 Verification is also performed.
`*SYS:<SYS:/V/S/T/H` ↓	Compress the SYS: device to remove any embedded empty files that may be present.

NON-DESTRUCTIVE FILE TRANSFERS

NOTES FOR /N

- As files are transferred, the device directory is updated to include all files present on the Output device. Input device files with different names are added to any Output device files that may be present. Files on the Input device replace files on the Output device only if the file names are identical.

- Files are always transferred in the order that they appear in the Input device directory (from beginning to end).

- If /D is specified, old files of the same name are deleted before the transfer. Files are then placed in the "best fit" holes available (usually in the same slots as the files that they are replacing).

- If /D is not specified, old files of the same name are not deleted until after the transfer. Files are placed in the "largest empty" holes available (usually at the end of the device). All transferred files will tend to remain grouped together.

`*DSK1:<DSK2:/V/N` ↓	Transfer all files from DSK2: to DSK1:. Files are transferred in DSK2:'s directory order, and are placed in the "largest empty" holes available (usually at the end of DSK1:.
`*DSK1:<DSK2:NAME1/V/N` ↓	Transfer files from DSK2: to DSK1: starting with the file NAME1. Files listed in the DSK2: directory before NAME1 are not transferred.
`*DSK1:<DSK2:NAME1,NAME2/V/N/O` ↓	Transfer only the files NAME1 and NAME2 from DSK2: to DSK1:. Files are placed in the "largest empty" holes available.
`*DSK1:<DSK2:/V/N/D` ↓	Transfer all files from DSK2: to DSK1: but insert them individually into the "best fit" holes available.
`*DSK1:<DSK2:NAME1,NAME2/V/N/O/D` ↓	Transfer only files NAME1 and NAME2 from DSK2: to DSK1:. Files are inserted into the "best fit" holes available.
`*LPT:<DSK2:NAME1/V/N/O/H` ↓	Output the file NAME1 on DSK2: to the Line Printer. Pass 2 (and T) Verifications must be inhibited for a transfer to a write only device.

3.2.5.6 VPIP Error Message & Warnings

The following VPIP error messages and warnings may occur:

Message	Description
ARE YOUR SURE?	A destructive /S transfer has been specified. A user must answer Y (Yes) to proceed. Typing N (No) or any other character returns control to the command decoder.
CANT COMPARE FILE (NAME)	Previously reported I/O errors have occurred while trying to read from either an Input or Output device during a Pass 2 Verification. Verification continues with the next file.
CANT FIND FILE (NAME)	The file named was not located on the Output device directory during verification. Verification continues with the next file.
CANT READ FILE (NAME)	Previously reported I/O errors have occurred while trying to read the file named from the Input device while using /T during Pass 1. Verification continues with the next file.
CANT READ INPUT DEVICE DIRECTORY	Previously reported I/O errors have occurred while trying to read the Input device directory before attempting a transfer from it. A transfer is not attempted, and control returns to the command decoder.
CANT READ OUTPUT DEVICE!	Previously reported I/O errors have occurred while trying to read the Output device directory before attempting a destructive /S transfer to it. A transfer is not attempted and control returns to the command decoder.
CANT WRITE OUTPUT DEVICE DIRECTORY	Previously reported I/O errors have occurred while trying to write the output device directory during a destructive /S transfer. Output is suspended, and control returns to the command decoder.
CANT WRITE FILE (NAME)	Previously reported I/O errors have occurred while trying to transfer the file named to the Output device while using /T during Pass 1. Verification continues with the next file.
DIF LNGTHD FILE (NAME)	The file named has a different length on the Output device than the file of the same name does on the Input device.

Message	Description
XX ERRORS	This is the final diagnostic result after completing Pass 2, where XX is the decimal count of the number of files failing Pass 2 Verification (can't find, different lengths, non-identical). Control returns to the command decoder.
NO ERRORS	No errors have been detected during the verification pass.
I/O ERROR $\left\{\begin{array}{ll}\text{WRITING} & \text{INPUT DEVICE \#n}\\ \text{READING} & \text{OUTPUT DEVICE}\end{array}\right\}$;nnn BLOCKS AT mmm	
Where:	
n	Specifies the Input device specification in the command decoder command string (1 to 9). For example: $\texttt{*DSK1:<}\underbrace{\texttt{DSK2:NAMEA}}_{1}\texttt{.}\underbrace{\texttt{DSK3:NAMEB}}_{2}\texttt{/V/N/O}$
nnn	Is the total number of octal blocks in the attempted transfer with errors.
mmm	Is the first physical octal block number in the attempted I/O transfer with errors. I/O transfers are attempted three times before the operation is aborted.
MONITOR ERROR 2 (or 6) during a VPIP /V/N transfer operation.	These are system errors that can occur while attempting to read or write a directory block (2) or if a directory overflow has occurred (6). If either error occurs while performing a /V/N transfer operation, the system date may be incorrect. Date the system again. For example, 1-APR-79 ↓ dates directory files (DAY-MONTH-YEAR)
NEED OUTPUT DEVICE	No Output device has been listed in the command decoder command line. Returns control to the command decoder.

Message	Description
NO DIRECTORY ROOM FOR (FILE-NAME)	During a destructive /S transfer, room in the directory has been exhausted and the (FILE-NAME) cannot be entered. The transfer is terminated (that is, the directory is written to reflect only those files that were successfully transferred) and verification, if any, starts.
NO ROOM ON OUTPUT DEVICE FOR (FILE-NAME)	The (FILE-NAME) will not fit on the Output device during a /N or /S transfer. The transfer is terminated (that is, the directory is written to reflect only those files that were successfully transferred) and verification, if any, starts.
NON-IDNTCL (FILE-NAME)	The (FILE-NAME) on the Output device was similar in size but different in content from the file of the same name on the Input device.
GR-VPIP VERSION XX	Invoked by the /W Option, where XX is the version number of the VPIP Program in use.

3.2.6 GenRad Archive Facility

3.2.6.1 Introduction

The GenRad Archive Facility extends the directory capability of an OS/8 file-structured device. This is accomplished by the creation of a "psuedo-device" within an OS/8 file-structured device. A psuedo-device is a file that has the characteristics of a file-structured device, containing both a directory and a file storage space of its own. The psuedo-device is referred to as an archive file. Thus, many files can be stored and accessed as necessary under a single file name. More than one archive file can exist. The number of files is limited only by the number of directory entries permitted for the OS/8 structured device (approx. 250).

3.2.6.2 Structure

The Archive Facility consists of two components: a device handler called ARCH: and an Archive Support program referred to as ASP. The ARCH: handler is normally inactive; that is, cannot be used without first running ASP. Attempts to use ARCH: for reading and writing while it is inactive will return errors. Additionally, when the handler is active, it must be specifically enabled for writing, or attempts to write using ARCH: will also return errors.

ASP ensures a high level of protection to an OS/8 device and any archive files resident on it by restricting activities that can be performed to archive files. ASP uses the Digital Equipment Corp. OS/8 FOTP program to copy, delete and rename files in an archive. The OS/8 DIRECT program is used to obtain an archive directory. ASP also provides for the creation of new archive files, and the Zero and Squish functions of existing ones.

3.2.6.3 Using The Archive Facility

The options listed and described below are used either when a new archive file is created, or when an existing file is accessed. The archive support program ASP must be run prior to initiating either type of operation (.R ASP).

<u>Creating A New Archive File</u> - An archive file is created when ASP is run by responding to the command decoder prompt (*) as follows:

```
*DEV:NAME.AR<= n(options)
```

ASP creates an archive file named NAME.AR on the device DEV:. The n specifies the number of blocks, in <u>octal</u>, that the archive file will contain. Of this number, the first seven blocks are used as the archive directory, and the remainder is for file space. At least 33 octal blocks <u>must</u> be specified for n as the <u>minimum</u> size archive file that ASP can create. Once an archive file exists it can be expanded or contracted to 27 or more decimal blocks. Additionally, an archive file can optionally be transferred to the same or another device, retaining or changing the name of the file, and the file length can be modified, all in one command.

Command options as necessary are specified following the n specification. These options are described in the next paragraph.

<u>Using An Existing Archive File</u> - Before any write-up operations (files added, zero or squish) take place the archive file must be write-enabled. With the command format:

```
*<DEV:NAME.AR (options)
```

ASP uses the archive file NAME.AR on the device DEV:. ASP also checks that the file specified is an archive file. The allowable options are:

/W Write enable the archive file. If this option is not present, /Z, /S and any use of ARCH: as an output device for the programs FOTP and DIRECT will fail.

/Z Zero the archive directory. When a new archive file is created it is not necessary to zero the directory. (In order to zero a directory the /W option is req.)

/S Squish. Either the embedded empty blocks within the archive file are consolidated while the archive file length remains fixed at its previously specified size, or the length of the archive can be compressed to a minimum size. When a squish is performed within an archive, the embedded empty blocks are placed at the end of the archive area. To perform this type of squish a command of the general form:

```
*name.AR/S/W
```

is used. Note that the /W option is required along with the /S here.

If the length of an existing archive file is to be changed, and the new length specified is smaller than the number of blocks already being used, the archive will be of such a size that there is no empty space. Similarly, if /S= is specified as either /S=0 or /S, then the length will shrink to the minimum size possible. A command of the form:

```
*nameb.AR<namea.AR/S/W
```

results in an archive of minimum size.

/F Chain to FOTP and execute exactly one command line. ASP types FOTP: and waits for a user-command line. Any FOTP program options are specified at this time.

/D Chain to DIRECT and execute exactly one command line. ASP types DIR: and waits for a user-command line. The command line should include any options the user requires with DIRECT.

/V Chain to VPIP and execute exactly one command line. ASP types VPIP: and waits for a user command line. This allows the user to compare a single file with another or to verify the contents of an entire disk. This option can also copy a file or files from a remote device before verification. Archived files may also be verified in this manner.

The < symbol is a file-specification separator used as shown to identify the file as an input file. The use of the < symbol is optional, but recommended in this instance. If the symbol is inadvertently placed at the end of a command, it is interpreted as a command to create a new archive output file rather than a command to operate on an existing archive input file. If the practice of explicit placement of the < symbol is followed in commands which operate on existing files, any later typographical errors will not have detrimental effects on the archive file. The < symbol must be used as a separator when operating on existing archive files as shown in the following examples:

```
*ARCH:<DSK1:NAMEA.SV or
FOTP:*DSK1:<ARCH:NAMEA.PA
```

Examples - The following examples typify archive file operations:

Create a new archive file of 100 octal blocks called STORE.AR and copy save files into it from DSK1:.

```
.R ASP
*STORE.AR</W/F/S=100
FOTP:*ARCH:<DSK1:*.SV
```

List the directory of the archive file MACLIB. (This is the system macro library.)

```
.R ASP
*MACLIB.AR/D
DIR:*TEC:<ARCH:/E=2
```

Copy out of the archive file USER1.AR, all files with a .PA extension and transfer them to device DSK1:.

```
.R ASP
*USER1.AR/F
FOTP:*DSK1:<ARCH:*.PA
```

Copy the complete archive file USER1.AR. The file is transferred to DSK2:, creating another new archive file called USER2.AR, that will contain all of the USER1.AR files and no empty blocks. (With no S=n specification, the archive shrinks to the smallest possible size.)

```
.R ASP

.*DSK2:USER2.AR<DSK1:USER1.AR/W/S
```

The above examples can be combined and run in a batch job.

```
$JOB BATCH EXAMPL.BI
/CREATE AN ARCHIVE AND COPY SAVE FILES FROM DSK1:
.R ASP
*STORE.AR</W/F/S=100
*ARCH:<DSK1:*.SV
/LIST ARCHIVE DIRECTORY FOR MACLIB
.R ASP
*MACLIB.AR/D
*TEC:<ARCH:/E=2
/COPY .PA FILES FROM USER1.AR ARCHIVE TO DSK1:
.R ASP
*USER1.AR/F
*DSK1:<ARCH:*.PA
/COPY DSK1:USER1.AR TO DSK2: AND RENAME IT USER2.AR
.R ASP
*DSK2:USER2.AR<DSK1:USER1.AR/W/S
$MSG DONE
$END
```

Running this batch file will result in a TTY hard copy as follows:

```
$JOB BATCH EXAMPL.BI
/CREATE AN ARCHIVE AND COPY SAVE FILES FROM DSK1:
.R ASP
*STORE.AR</W/F/S=100
FOTP:*ARCH:<DSK1:*.SV
/LIST ARCHIVE DIRECTORY FOR MACLIB
.R ASP
*MACLIB.AR/D
DIR:*TEC:<ARCH:/E=2           (See note below)
/COPY .PA FILES FROM USER1.AR ARCHIVE TO DSK1:
.R ASP
*USER1.AR/F
FOTP:*DSK1:<ARCH:*.PA
/COPY DSK1:USER1.AR TO DSK2: AND RENAME IT USER2.AR
.R ASP
*DSK2:USER2.AR<DSK1:USER1.AR/W/S
$MSG DONE
$END
#END BATCH
```

> Note, that the batch job requests a TEC display for the MACLIB.AR directory. Batch processing will halt and will not resume until the TEC is cleared by the operator. If the TTY is selected the directory will be typed and batch processing will continue without operator intervention.

The following example illustrates how macros, in existing macro files, can be modified. This process may prove particularly useful with Analog Diagnostics, where the software is built using macro routines stored in ANALOG.AR.

Diplay the directory of the Analog Diagnostic file, ANALOG.AR.

```
.R ASP
*ANALOG.AR/D
DIR:*TEC:<ARCH:/E=3
```

Copy the Macro called PASPRB from the archive file ANALOG.AR to DSK1:, so that it can be looked at, listed or edited.

```
.R ASP
*ANALOG.AR/F
FOTP:*DSK1:<ARCH:PASPRB
```

Edit PASPRB following normal editing procedure with I=PASPRB; &
O=PASPRB;
(no extension is specified). Use LOOK mode to provide hard copy.

Put edited PASPRB macro back into ANALOG.AR archive.

```
.R ASP
*ANALOG.AR/W/F
FOTP:*ARCH:<DSK1:PASPRB
```

Check Directory; the macro that was returned to the archive file should now have a new date.

```
.R ASP
*ANALOG.AR/D
DIR:*TEC:<ARCH:/E=2
```

Assuming one is now done making modifications on PASPRB, and the corrected version has been put back into the archive file, one may now DELETE the PASPRB file from the working surface (DSK1: in this case) to conserve working space.

3.2.6.4 Error Messages And Conditions

Most of the error messages encountered from the use of the archive facility are self-explanatory. All messages appear below, along with comments as required.

? NO ARCHIVE HANDLER
ARCH: does not exist

? BAD ARCHIVE HANDLER
ARCH is not an archive handler. This message is most likely to occur if some other handler is assigned as ARCH:

? ARCH:/ASP MISMATCH
ASP is not compatible with the version of ARCH: on the system.

? SPECIFIED FILE IS NOT AN ARCHIVE FILE

? INSUFFICIENT ROOM FOR ARCHIVE FILE

? CAN`T USE DEVICE AS AUXILLARY HANDLER
The device was not file structured.

? AUXILLARY HANDLER MUST BE 1 PAGE

? NO FILENAME SPECIFIED
The command decoder line contained a device name but no filename.

? ARCHIVE FILE/ASP MISMATCH
ASP cannot deal with the specified archive file.

? SYSTEM PROGRAM NOT FOUND
A selected system program, FOTP or DIRECT was not on SYS:.

? NOT ENOUGH memory AVAILABLE
12K is required but is not available.

? ARCHIVE MUST BE WRITE ENABLED (/W) TO SQUISH

? SYSTEM I/O ERROR
ASP encountered an I/O error while reading from or writing to SYS:.

? ARCHIVE I/O ERROR
ASP encountered an I/O error while reading or writing the user`s archive file.

3.2.6.5 Storing User-Language Macros In An Archive File

User-language macros to be used in a test program(s) are stored in an archive file in the following manner:

a. Using the EDITor create a macro-file that has the .DEFINE and .ENDM statements at the beginning and end respectively, as described in the Programming Test Language Manual. The I= and O= editor parameters are given the name of the macro, exactly as specified in single quotes in the test program .DEFINE statement. No extension should be stated, just the macro name.

b. The completed macro file is then ready for transfer to the archive file. Do not run the macro file through the translator at this time. It will be processed when the rest of the test program is translated. Run ASP as described in this section, and transfer any macros into it.

3.2.6.6 Using Macros In An Archive File - An Example

The following example is illustrated to further clarify the operations involving the use of an archive file for storage of test program macros. In this example, three macros called M1, M2 and M3 are created as three different files using the Editor, as described in para. 3.2.6.5.

```
.DEFINE `M1`;  /FILE #1 NAME = M1/
/MACRO1/

IH(1);
IL(1);

/END MACRO1/

.ENDM;

.DEFINE `M2`;  /FILE #2 - NAME = M2/

/MACRO2/

IH(2);
IL(2);

/END MACRO2/

.ENDM;

.DEFINE `M3`;  /FILE #3 - NAME = M3/

/MACRO3/

IH(3);
IL(3);

/END MACRO3/

.ENDM;
```

The Archive Support Program (ASP) is then run from OS/8 and an archive file called USER1.AR is established with 100 decimal blocks (144 <u>octal</u> blocks) of storage, and the directory zeroed. The OS/8 program FOTP is then called in order to copy the macros into USER1.AR. The USER1.AR directory is verified using the commands shown.

```
.R ASP
*USER1.AR</W/Z=144            /OPEN A FILE 100 DECIMAL BLOCKS LONG &
.R ASP                        /ZERO THE DIRECTORY
*USER1/W/F                    /CHAIN TO FOTP
```

```
FOTP:*ARCH:<DSK1:M1,M2,M3     /COPY THE MACROS INTO THE ARCHIVE FILE

.R ASP                        /CHAIN TO THE OS/8 DIRECT PROGRAM
*USER1/D

DIR:*TEC:<ARCH:               /LOOK AT THE DIRECTORY


11-MAR-77

M2      .      1 11-MAR-77

M3      .      1 11-MAR-77
M1      .      1 11-MAR-77

  90 FREE BLOCKS
```

The following test program illustrates how macros that have been defined in an archive file are called out:

```
BEGIN CAPS `TRY`;

SET PS-1 V=5 I=3;

ASSIGN LGC IA(1-12) OA(13-24)
VOLA=1.5 VOHA=1.5 VILA=0 VIHA=3;

IL(#) OH(#);
.`M1`;
.`M2`;
.`M3`;

FINISH CAPS;
END;
```

A test program containing user-defined macros is translated as shown below. Note that the archive file USER1.AR is specified (LM =) and that the macro processor (M) and re-number (R) options are turned on. Additionally, if the translate P option is on, the expanded test program is printed as shown below.

When M is turned on and a macro is encountered, the software firsts checks to see if the macro is defined at the beginning of the test program. If the macro is not defined there, the software searches the archive file specified by the LM= option. If the macro is not in the specified user-language macro archive file, the software looks in the system MACLIB.AR library file. If the macro called for is not found within the system macro library file, an UNDEFINED MACRO error message is returned.

```
***** SAMPLE MONITOR PAGE SETUP *****

        FILE USAGE                                      TRANSLATE MODE
        -----------
TP =    DSK:MACRO.TP     TEST PROGRAM NAME
W  =    DSK2:, DSK3:     WORKING DISKS
X  =    DSK2:, DSK3:     EXTENDED WORKING DEVICES
LI =    TEC:             LISTING DEVICE
LM =    DSK:USER1.AR     USER LANGUAGE MACRO FILE


              TRANSLATE OPTIONS (COMMANDS: TY, TN, TO, TA)
              --------------------------------------------

OPTION  STATUS    DESCRIPTION
M       ON        RUN MACRO PROCESSOR
R       ON        RE-NUMBER PROGRAM STATEMENTS IN MACRO PROCESSOR
D       OFF       DELETE UNREFERENCED STATEMENT NUMBERS IN MACRO PROCESSOR
X       ON        PRINT MACO-EXPANDED TEST PROGRAM
P       ON        PRINT TEST PROGRAM SOURCE
E       ON        EDIT ON ERRORS IN TRANSLATION


/******* EXPANDED TEST PROGRAM LISTING *******/

BEGIN CAPS `TRY`;
SET PS-1 V=5 I=3;
ASSIGN LGC IA(1-12) OA(13-24)
VOLA=1.5 VOHA=1.5 VILA=0 VIHA=3;
IL(#) OH(#);
/MACRO1/

IH(1);
IL(1);

/END MACRO1/


/MACRO2/

IH(2);
IL(2);

/END MACRO2/


/MACRO3/

IH(3);
IL(3);

/END MACRO3/
FINISH CAPS;
END;
```

3.3 GenRad TEST, CAPS, AND APG MONITORS

3.3.1 General

The GenRad monitors (*) control and allow access to a series of major modes and utility programs in much the same manner as the OS/8 keyboard monitor controls and allows access to OS/8 system programs.

The TEST Monitor allows access to all of the non-CAPS/APG operating modes that would be required to both prepare the files needed for device testing and then test devices (EDIT, TRANSLATE, and DIAGNOSE). In addition, some special purpose modes are also supplied to allow efficient file maintenance and perform other special functions (CLEANUP, LOOK, CONVERT, BATCH, and SETUP).

The CAPS and APG Monitors allow access to all of the CAPS modes (PREPARE, GENERATE, DISPLAY, LIBLIST, LISTALL, ICDELETE, ICINFO, MAKE, LOG, NODALV and 2225 mode addition to the non-CAPS modes just described (see Figure 3-2).

In addition the APG Monitor allows access to the E/T/C Mode and Incremental Simulation in the PREPARE Mode.

The following paragraphs define how to access each monitor, access each operating mode from that monitor, issue intra-mode commands, and use some of the special purpose features available such as positional arguments, macro files, and batch job runs. A complete description of the CAPS and APG operating modes is contained in the CAPS/APG Operating Instructions. Detailed descriptions of the non-CAPS/APG modes (EDIT and TRANSLATE) and the CONVERT Mode are located in the standard test system operating instructions. Some modes (such as DIAGNOSE) operate in a slightly different manner when used with each monitor because the non-CAPS/APG mode either does not have the CAPS commands available to it or uses the commands in a different manner. These modes (DIAGNOSE, CLEANUP, and BATCH), are described in both publications.

3.3.2 Accessing The TEST, CAPS And APG Monitors

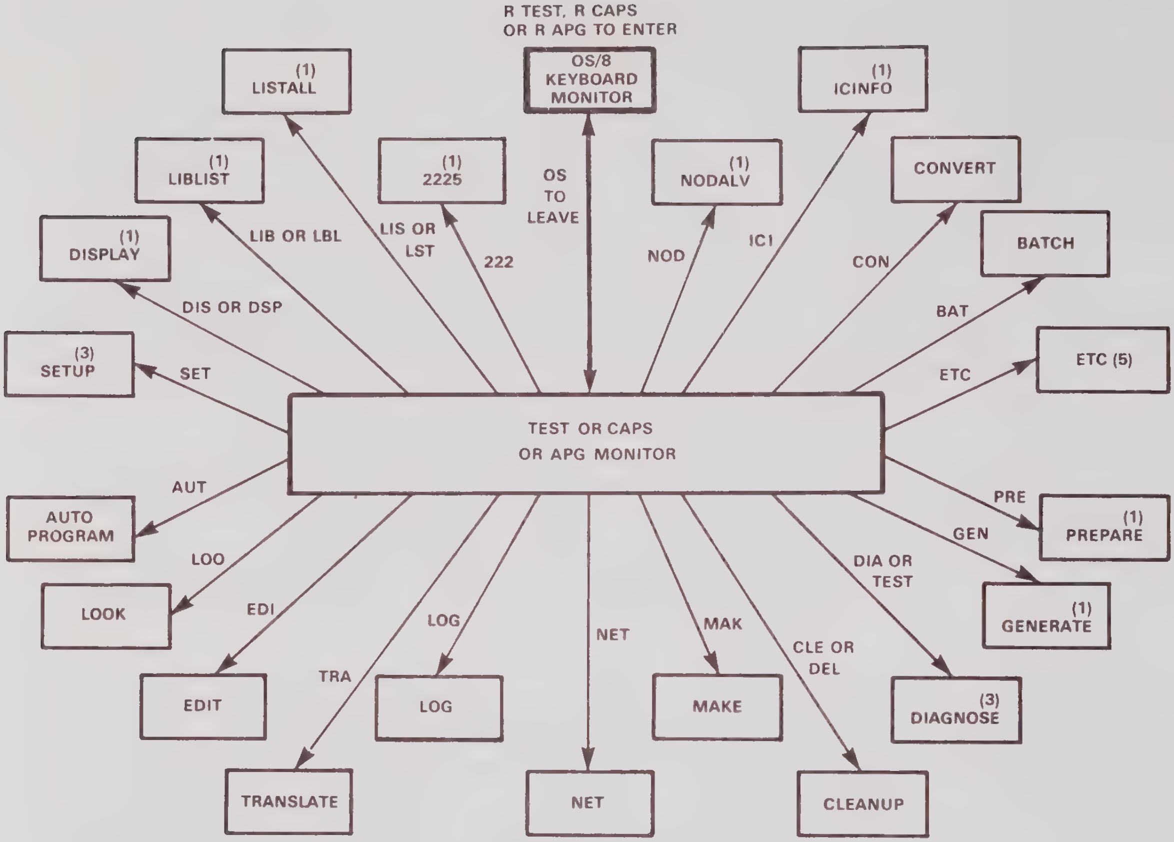
Each of the monitors is accessed from the OS/8 Keyboard Monitor by entering the appropriate command through the Teletype keyboard. For example,

```
        .R TEST ⤶                To call the TEST Monitor

          or

        .R CAPS ⤶                To call the CAPS Monitor

          or

        .R APG ⤶                 To call the APG Monitor
```

Note that underlined characters are echoed by Teletype.
The (⤶) character indicates the Teletype RETURN key.

---

* A single CAPS, TEST, or APG Monitor is furnished.



R TEST, R CAPS OR R APG TO ENTER
OS/8 KEYBOARD MONITOR
OS TO LEAVE
(1) LISTALL
(1) LIBLIST
(1) 2225
(1) NODALV
(1) ICINFO
CONVERT
BATCH
(1) DISPLAY
LIB OR LBL
LIS OR LST
222
NOD
ICI
CON
BAT
DIS OR DSP
(3) SETUP
SET
ETC
ETC (5)
TEST OR CAPS OR APG MONITOR
AUT
AUTO PROGRAM
PRE
(1) PREPARE
GEN
LOO
LOOK
DIA OR TEST
(1) GENERATE
EDI
LOG
NET
MAK
CLE OR DEL
EDIT
TRA
LOG
MAKE
(3) DIAGNOSE
TRANSLATE
NET
CLEANUP

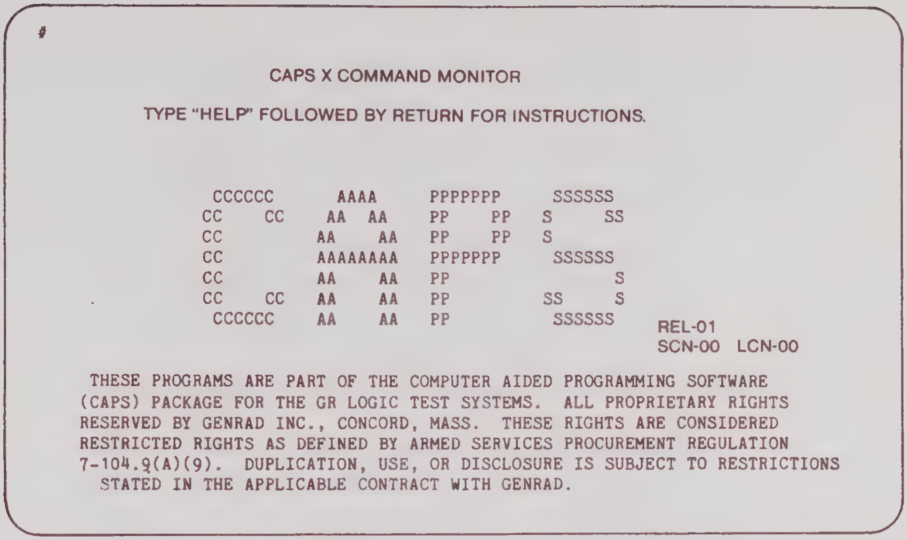
NOTES

(1) SUPPLIED WITH CAPS AND APG SYSTEMS ONLY.
(2) USE EXIT OR CTRL C TO RETURN TO MONITOR FROM ANY MODE OR PROGRAM. USE OS TO RETURN TO OS/8.
(3) CAPS AND APG MODES ARE NOT SUPPLIED WITH NON-CAPS VERSIONS OF A MODE OR PROGRAM.
(4) ONLY THE FIRST THREE CHARACTERS OF EACH COMMAND ARE NEEDED TO ENTER A MODE OR PROGRAM.
(5) SUPPLIED WITH APG ONLY.

Figure 3-2 Methods of entering modes and programs controlled by a GR monitor.

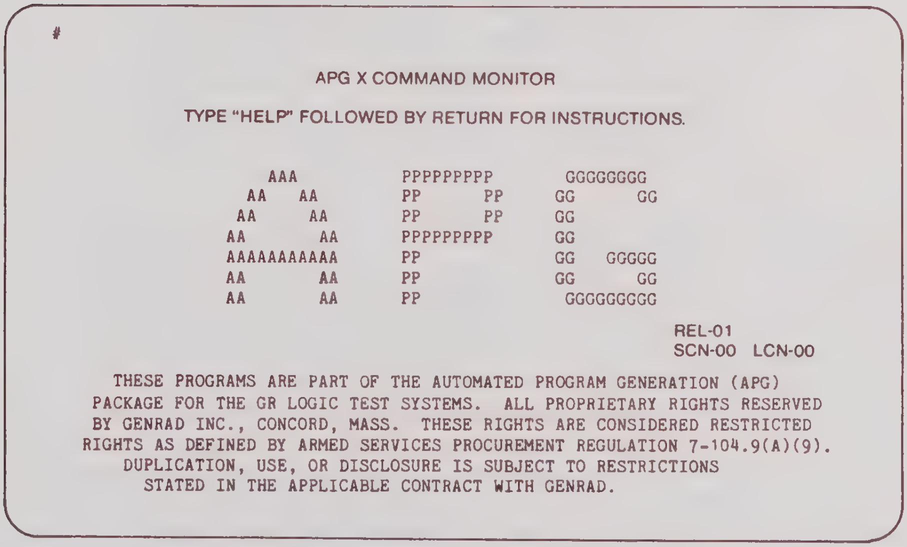
When the GenRad Monitor selected is in memory and ready for use, a top level page will be displayed on the TEC data terminal screen (see Figure 3-3). The top level page identifies the monitor being used (CAPS, APG, etc.) and also lists the software release involved (REL-XX). The user may then enter commands via the Teletype or data terminal keyboards. Commands entered through the Teletype keyboard are echoed by Teletype in addition to being displayed in the upper left corner of the display screen. Thus, hard copies of each command that has been issued will be available for future reference, if desired. Table 3-3 defines some of the special keys that may be used to enter or correct command lines specified by Teletype, when either GenRad Monitor is in use.

Command lines may also be entered through the data terminal keyboard and will be displayed in the upper left corner of the display screen as they are entered. Any of the data terminal keyboard editing keys may be used to correct command lines. The only restrictions are that commands may not be started on one keyboard and finished on another. Select one of the keyboards and enter a complete command line before changing keyboards.



```
#

                    CAPS X COMMAND MONITOR

          TYPE "HELP" FOLLOWED BY RETURN FOR INSTRUCTIONS.


             CCCCCC      AAAA     PPPPPPP     SSSSSS
            CC    CC    AA  AA    PP    PP   S     SS
            CC         AA    AA   PP    PP   S
            CC         AAAAAAAA   PPPPPPP     SSSSSS
            CC         AA    AA   PP                S
            CC    CC   AA    AA   PP         SS     S
             CCCCCC    AA    AA   PP          SSSSSS
                                                        REL-01
                                                        SCN-00  LCN-00

  THESE PROGRAMS ARE PART OF THE COMPUTER AIDED PROGRAMMING SOFTWARE
 (CAPS) PACKAGE FOR THE GR LOGIC TEST SYSTEMS.  ALL PROPRIETARY RIGHTS
 RESERVED BY GENRAD INC., CONCORD, MASS.  THESE RIGHTS ARE CONSIDERED
 RESTRICTED RIGHTS AS DEFINED BY ARMED SERVICES PROCUREMENT REGULATION
 7-104.9(A)(9).  DUPLICATION, USE, OR DISCLOSURE IS SUBJECT TO RESTRICTIONS
   STATED IN THE APPLICABLE CONTRACT WITH GENRAD.
```

CAPS MONITOR TOP LEVEL PAGE DISPLAY



```
#

                      APG X COMMAND MONITOR

            TYPE "HELP" FOLLOWED BY RETURN FOR INSTRUCTIONS.


                    AAA          PPPPPPPPP       GGGGGGGG
                  AA   AA        PP      PP     GG      GG
                 AA     AA       PP      PP     GG
                AA       AA      PPPPPPPPP      GG
                AAAAAAAAAAA      PP             GG   GGGGG
                AA       AA      PP             GG      GG
                AA       AA      PP              GGGGGGGGG
                                                            REL-01
                                                            SCN-00  LCN-00

    THESE PROGRAMS ARE PART OF THE AUTOMATED PROGRAM GENERATION (APG)
  PACKAGE FOR THE GR LOGIC TEST SYSTEMS.  ALL PROPRIETARY RIGHTS RESERVED
  BY GENRAD INC., CONCORD, MASS.  THESE RIGHTS ARE CONSIDERED RESTRICTED
 RIGHTS AS DEFINED BY ARMED SERVICES PROCUREMENT REGULATION 7-104.9(A)(9).
     DUPLICATION, USE, OR DISCLOSURE IS SUBJECT TO RESTRICTIONS
       STATED IN THE APPLICABLE CONTRACT WITH GENRAD.
```

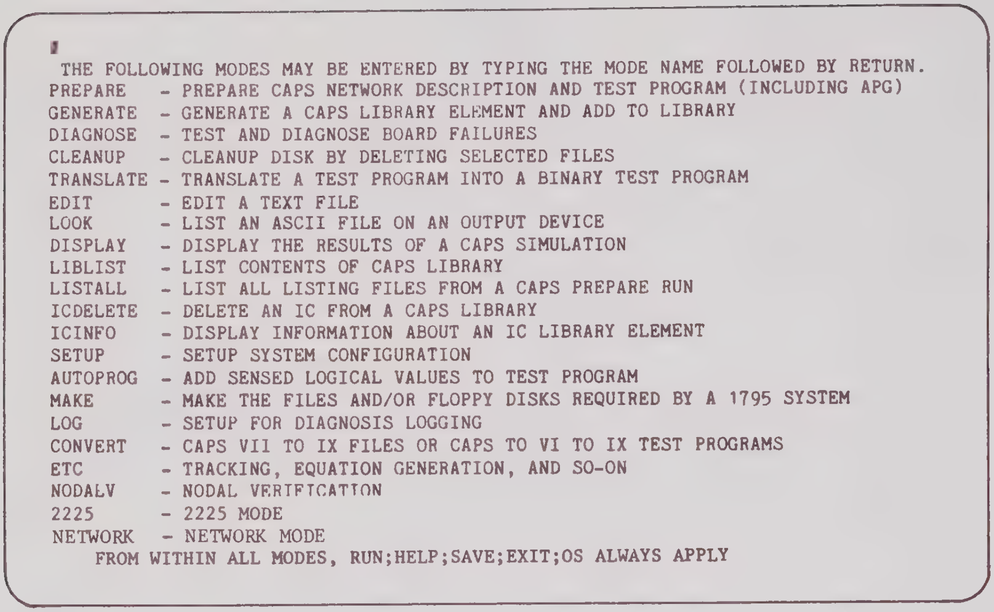
Figure 3-3. CAPS/APG Monitor Top Level Page Displays.

Table 3-3

SPECIAL TELETYPE KEY FUNCTIONS
AVAILABLE WITH ANY GR MONITOR

Key	Function
LINE FEED	Echoes back the input line just entered.
RUBOUT	Deletes the last character displayed.
CTRL U(1)	Deletes the complete command line. A new command line may now be entered.
RETURN(2)	Listed in text as (↓). Executes the command line specified.

(1) Hold down the CTRL key while pressing the letter key indicated.
(2) The ALTMODE or ESC keys can also be used to enter a command line.



```
 THE FOLLOWING MODES MAY BE ENTERED BY TYPING THE MODE NAME FOLLOWED BY RETURN.
PREPARE   - PREPARE CAPS NETWORK DESCRIPTION AND TEST PROGRAM (INCLUDING APG)
GENERATE  - GENERATE A CAPS LIBRARY ELEMENT AND ADD TO LIBRARY
DIAGNOSE  - TEST AND DIAGNOSE BOARD FAILURES
CLEANUP   - CLEANUP DISK BY DELETING SELECTED FILES
TRANSLATE - TRANSLATE A TEST PROGRAM INTO A BINARY TEST PROGRAM
EDIT      - EDIT A TEXT FILE
LOOK      - LIST AN ASCII FILE ON AN OUTPUT DEVICE
DISPLAY   - DISPLAY THE RESULTS OF A CAPS SIMULATION
LIBLIST   - LIST CONTENTS OF CAPS LIBRARY
LISTALL   - LIST ALL LISTING FILES FROM A CAPS PREPARE RUN
ICDELETE  - DELETE AN IC FROM A CAPS LIBRARY
ICINFO    - DISPLAY INFORMATION ABOUT AN IC LIBRARY ELEMENT
SETUP     - SETUP SYSTEM CONFIGURATION
AUTOPROG  - ADD SENSED LOGICAL VALUES TO TEST PROGRAM
MAKE      - MAKE THE FILES AND/OR FLOPPY DISKS REQUIRED BY A 1795 SYSTEM
LOG       - SETUP FOR DIAGNOSIS LOGGING
CONVERT   - CAPS VII TO IX FILES OR CAPS TO VI TO IX TEST PROGRAMS
ETC       - TRACKING, EQUATION GENERATION, AND SO-ON
NODALV    - NODAL VERIFICATION
2225      - 2225 MODE
NETWORK   - NETWORK MODE
    FROM WITHIN ALL MODES, RUN;HELP;SAVE;EXIT;OS ALWAYS APPLY
```

THE APG MONITOR HELP PAGE. THE CAPS
HELP PAGE IS SIMILAR, BUT NOT IDENTICAL.

Figure 3-4. A Typical Top Level Monitor HELP Page.

All commands entered through the data terminal keyboard are activated by depressing the XMIT key after making sure that the display cursor is at least one space to the right of the command string (all characters entered to the left of the cursor are considered to be part of the command string to be executed). For example, the command string.

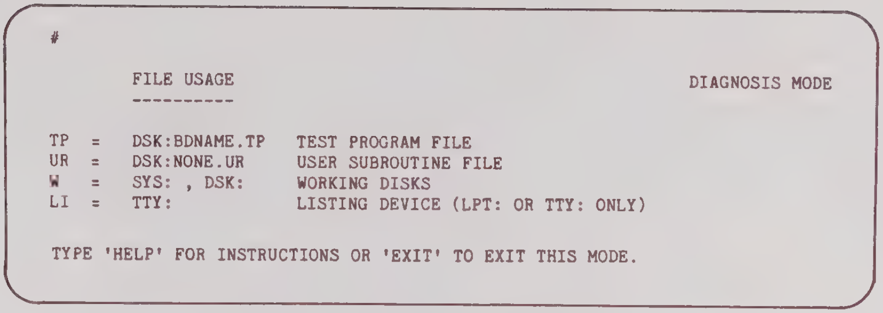
```
         ┌CURSOR
#HELP _↙         and XMIT or RETURN
```

will result in a display of the HELP page associated with the monitor top level (see Figure 3-4). Note that the (#) character display identifies the start of a monitor command line and is not entered by the user. If commands are issued from both keyboards (or if the displayed # is erased), the command line will not be executed when the XMIT (or RETURN) key is depressed; instead, a NO SUCH COMMAND message will be generated, and a new # character will be automatically added so that the correct command line can be entered.

The top level HELP page defines the operating modes available within each monitor. To enter an operating mode, it is only necessary to type the first three characters of the name assigned to that mode (although the complete name may be entered). For example,

```
# EDI     and XMIT or RETURN
```

will cause the EDIT mode to be entered. It should be noted that the top level HELP page is equivalent in every respect to the TEST, CAPS, or APG top level page, and therefore monitor commands may be issued directly from these pages. For example, the EDIT command just described can be issued with either page displayed, and will result in the EDIT mode being entered.



```
#

        FILE USAGE                                             DIAGNOSIS MODE
        ----------

TP  =   DSK:BDNAME.TP   TEST PROGRAM FILE
UR  =   DSK:NONE.UR     USER SUBROUTINE FILE
W   =   SYS: , DSK:     WORKING DISKS
LI  =   TTY:            LISTING DEVICE (LPT: OR TTY: ONLY)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 3-5. First Page of DIAGNOSE Mode.

To return to the GR Monitor top level (TEST, CAPS, or APG display) from within any operating mode (such as EDIT, TRANSLATE, etc.) enter one of the following commands:

```
# EXIT     and XMIT or RETURN

     or

  CTRL C        from the Teletype keyboard
```

To return to the OS/8 Keyboard Monitor from the top level of a GenRad Monitor or from any of the operating modes controlled by a GenRad Monitor, enter the following command:

```
# OS      and XMIT or RETURN
```

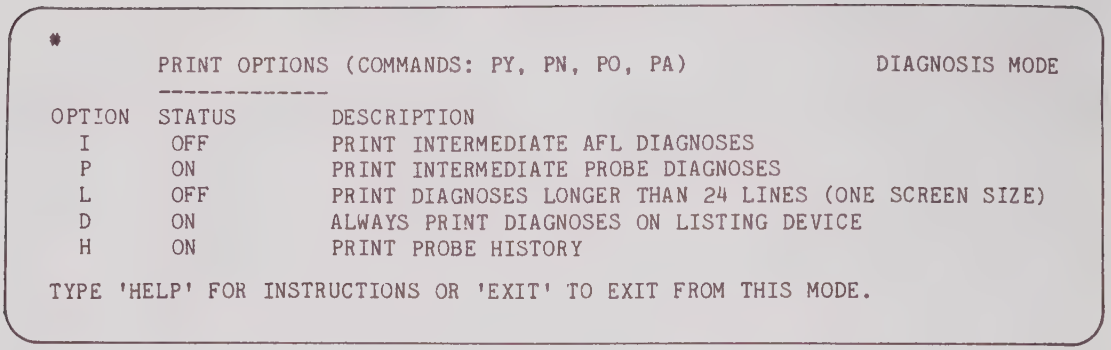
Since each command string is activated in the same manner (by depressing the data terminal XMIT or Teletype RETURN keys), only the command string itself will generally be listed in further references within this publication. It is assumed that the reader understands that each command line must be activated by depressing the keys listed above.

3.3.3 Accessing Monitor Operating Modes

Each operating mode may be entered by typing at least the first three characters of the name assigned to that mode. For example,

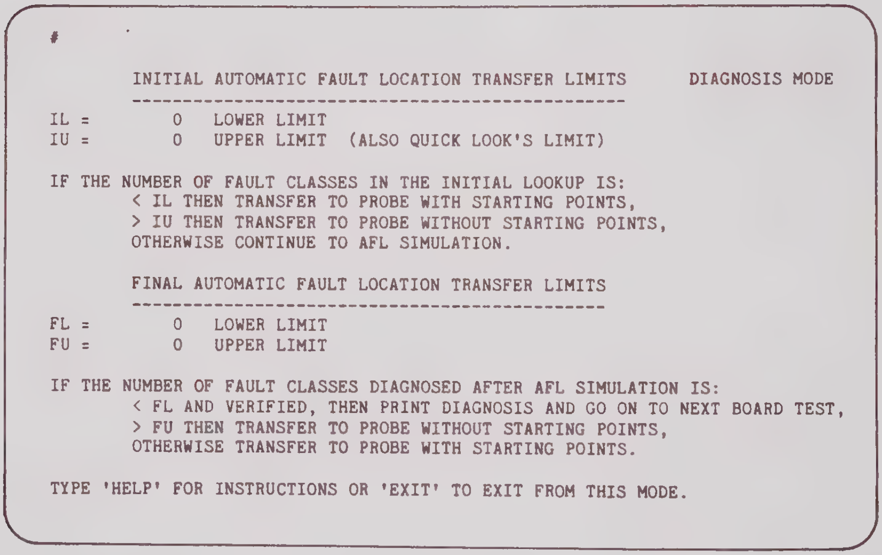
```
# DIA
```

will allow the DIAGNOSE Mode to be entered and the page shown in Figure 3-5 to be displayed. Each operating mode will display the first page of that mode when it is entered, and can be identified further by a flashing display of the mode name in the upper right corner of the screen. Each mode will contain at least one page for display and may have a series of pages. To view any additional pages within a mode, depress the data terminal XMIT (or Teletype RETURN) key. For example, depressing the XMIT key three times will result in a display of the next three pages in order. The last two of these pages are shown in Figure 3-6. Continuing to depress the XMIT key will continue to advance and display additional pages (if any) until the first page is again displayed.



```
*
        PRINT OPTIONS (COMMANDS: PY, PN, PO, PA)          DIAGNOSIS MODE
        -------------
OPTION  STATUS        DESCRIPTION
  I      OFF          PRINT INTERMEDIATE AFL DIAGNOSES
  P      ON           PRINT INTERMEDIATE PROBE DIAGNOSES
  L      OFF          PRINT DIAGNOSES LONGER THAN 24 LINES (ONE SCREEN SIZE)
  D      ON           ALWAYS PRINT DIAGNOSES ON LISTING DEVICE
  H      ON           PRINT PROBE HISTORY

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

THIRD PAGE



```
#


        INITIAL AUTOMATIC FAULT LOCATION TRANSFER LIMITS      DIAGNOSIS MODE
        ------------------------------------------------
IL =         0   LOWER LIMIT
IU =         0   UPPER LIMIT  (ALSO QUICK LOOK'S LIMIT)

IF THE NUMBER OF FAULT CLASSES IN THE INITIAL LOOKUP IS:
        < IL THEN TRANSFER TO PROBE WITH STARTING POINTS,
        > IU THEN TRANSFER TO PROBE WITHOUT STARTING POINTS,
        OTHERWISE CONTINUE TO AFL SIMULATION.

        FINAL AUTOMATIC FAULT LOCATION TRANSFER LIMITS
        ----------------------------------------------
FL =         0   LOWER LIMIT
FU =         0   UPPER LIMIT

IF THE NUMBER OF FAULT CLASSES DIAGNOSED AFTER AFL SIMULATION IS:
        < FL AND VERIFIED, THEN PRINT DIAGNOSIS AND GO ON TO NEXT BOARD TEST,
        > FU THEN TRANSFER TO PROBE WITHOUT STARTING POINTS,
        OTHERWISE TRANSFER TO PROBE WITH STARTING POINTS.

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

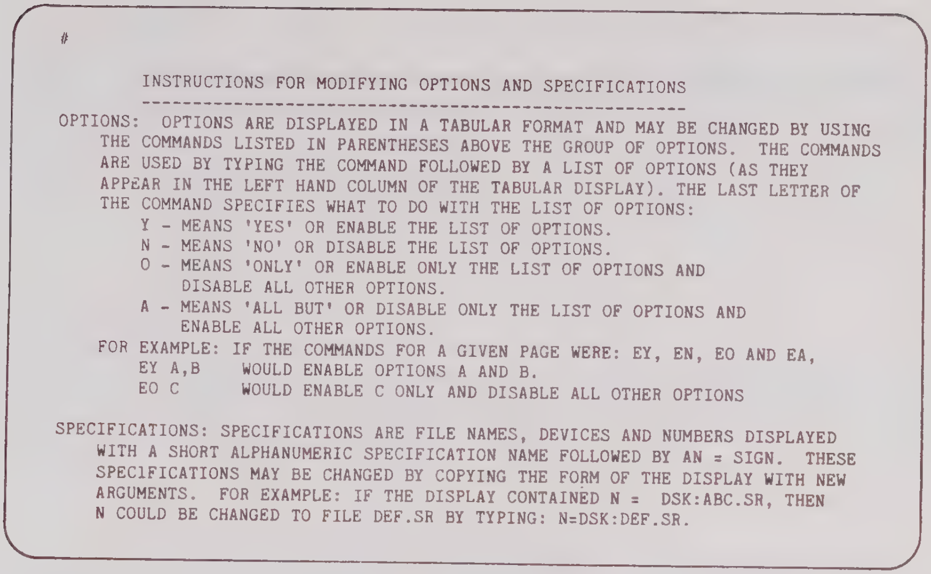
FOURTH PAGE

Figure 3-6. Third and Fourth Pages of DIAGNOSE Mode.

Each page contains a list of optional commands and/or file names that can be used by the mode in use. Optional commands and file names may be changed by typing commands to the monitor while viewing each page. For example, the test program file DSK:SAMPLE.TP listed beside the TP Option in Figure 3-5 may be changed to read DSK2:BOARDA.TP in the following manner:

```
#TP=DSK2:BOARDA.TP
```

When the XMIT key is depressed, the TP file name will be updated to read DSK2:BOARDA.TP. Once a change has been made to the page being displayed it will remain in effect until it is changed again. Even if the test system is shut down and then re-started, the new TP file name will remain in effect. Changes are permanently remembered after a RUN, EXIT, CTRL C, or OS command has been issued.



```
#

         INSTRUCTIONS FOR MODIFYING OPTIONS AND SPECIFICATIONS
         -----------------------------------------------------
OPTIONS:  OPTIONS ARE DISPLAYED IN A TABULAR FORMAT AND MAY BE CHANGED BY USING
    THE COMMANDS LISTED IN PARENTHESES ABOVE THE GROUP OF OPTIONS.  THE COMMANDS
    ARE USED BY TYPING THE COMMAND FOLLOWED BY A LIST OF OPTIONS (AS THEY
    APPEAR IN THE LEFT HAND COLUMN OF THE TABULAR DISPLAY). THE LAST LETTER OF
    THE COMMAND SPECIFIES WHAT TO DO WITH THE LIST OF OPTIONS:
        Y - MEANS 'YES' OR ENABLE THE LIST OF OPTIONS.
        N - MEANS 'NO' OR DISABLE THE LIST OF OPTIONS.
        O - MEANS 'ONLY' OR ENABLE ONLY THE LIST OF OPTIONS AND
            DISABLE ALL OTHER OPTIONS.
        A - MEANS 'ALL BUT' OR DISABLE ONLY THE LIST OF OPTIONS AND
            ENABLE ALL OTHER OPTIONS.
    FOR EXAMPLE: IF THE COMMANDS FOR A GIVEN PAGE WERE: EY, EN, EO AND EA,
        EY A,B     WOULD ENABLE OPTIONS A AND B.
        EO C       WOULD ENABLE C ONLY AND DISABLE ALL OTHER OPTIONS

SPECIFICATIONS: SPECIFICATIONS ARE FILE NAMES, DEVICES AND NUMBERS DISPLAYED
    WITH A SHORT ALPHANUMERIC SPECIFICATION NAME FOLLOWED BY AN = SIGN.  THESE
    SPECIFICATIONS MAY BE CHANGED BY COPYING THE FORM OF THE DISPLAY WITH NEW
    ARGUMENTS.  FOR EXAMPLE: IF THE DISPLAY CONTAINED N =  DSK:ABC.SR, THEN
    N COULD BE CHANGED TO FILE DEF.SR BY TYPING: N=DSK:DEF.SR.
```

NOTE

To return to the mode page from which the HELP command was issued, depress the XMIT or RETURN keys.

Figure 3-7. The Sub-Mode HELP Page.

Any number of options may be changed by the same command string if each option is separated by a semi-colon (;) as in the following example:

```
# IL=4;IU=6
```

The options listed are contained on the fourth page of the DIAGNOSE Mode shown in Figure 3-6. The IL= and IU= options are used to set Automatic Fault Location Parameters. In addition, the IU= option also sets the Quick Lookup transfer limits. Note that only options on the page being displayed can be changed from that page. Any option on any page within a mode may be changed from the GR Monitor top level by a single command string, however.

A HELP page may be accessed from any page within a mode by entering the command

```
# HELP
```

which invokes the HELP page shown in Figure 3-7. Commands may only be issued from the HELP page available at the monitor top level. Commands may not be issued from HELP pages called from within a mode. Depressing the XMIT or RETURN keys will result in a return to the mode page from which the HELP command was issued.

3.3.4 Intra-Mode Commands

There are three general types of commands that can be issued from within any mode (equate commands, option ON/OFF commands, and auxiliary commands). The following paragraphs define how to change each type of command.

3.3.4.1 Equate Commands (=Specifications)

Equate commands consist of one to four character identifiers always followed by an equal sign (see the fourth page display of Figure 3-6). Equate commands are used to specify file names, devices, special features, numbers within a mode, etc., and may be changed whenever the page involved is displayed. For example, the IL specification for Automatic Fault Location Parameters shown in Figure 3-6 may be changed from IL=4 to IL=3 in the following manner:

```
#IL=3
```

When the XMIT key is depressed, the new specification will be included beside the IL entry. Any of the data terminal keyboard editing keys can be used to correct command lines if erroneous data is entered. Once the command line has been corrected, and the cursor is at least one space to the right of the last command character, the XMIT key can be depressed to change the equate. Typical equate commands have the form:

B=n	Where n is a decimal number.
R=NAME	Such as Restart on $\underbrace{\text{PASS, FAIL, ALWAYS}}_{\text{Select one NAME}}$
T=DEV:NAME	Where the NAME and location of the file to be used is specified (for example, T=SYS:GRDEM.TP).
LI=DEV:	Where DEV: may be any Output Device (TTY:, TEC:, LPT:).

3.3.4.2 Option ON/OFF Commands

Option ON/OFF commands turn options ON or OFF within each mode. For example, the PRINT options available within the DIAGNOSE Mode are shown in Figure 3-6. The left hand column of each ON/OFF option list consists of one or two character option identifiers (I,L,D,H,etc.) followed by the current STATUS (ON or OFF) of each option, followed by a brief description of that option. Option identifiers serve as abbreviations for the options listed.

Commands that can be used to change options are included in parenthesis at the top of the page being displayed (PU, PN, PO, PA). The first character (P) identifies the type of option ON/OFF commands involved (P for Print, D for Diagnostic Modes, G for Guided Probe, etc.). The second character (Y,N,O,A) determines how the options listed are to be turned ON or OFF. Each of the characters have the following unique meaning:

Y Character (Yes)

Y - Turn ON the options listed without affecting any other options. For example, the I,L, and D options listed in Figure 3-6 could be turned ON in the following manner:

```
# PY I,L,D     or # PY I L D
```

Note that options not listed are not affected and remain as they were. At least one space must occur immediately after the PY command, and either a comma and/or one or more spaces must separate each option listed.

N character (No)

N - Turn OFF the options listed without affecting any other options. Note that N is the reverse of Y. For example, the I, L, and D Options can be turned OFF in the following manner:

```
# PN I,L,D
```

Options not listed are not affected and remain as they were.

O Character (Only)

O - Turn OFF all options, then turn ON only the options specified. For example, the I and P options can be turned ON while turning OFF all other options in the following manner:

```
# PO I,P
```

Note that the I and P options are set ON while all other options (L,D,H) are turned OFF. If the command is issued by itself (without specifying a list of options) all options will be turned OFF. For example,

```
# PO
```

will turn OFF all options (I,P,L,D, and H).

A Character (All But)

A - Turn ON all options, then turn OFF the options specified (turn ON all but the options listed). Note that A is the reverse of O. For example, the I and P options can be turned OFF while turning ON all other options in the following manner:

```
# PA I,P
```

If the command is issued without listing a set of options, all options will be turned ON. For example, the command

```
# PA
```

will turn ON all options (I,P,L,D, and H).

3.3.4.3 Auxiliary Commands

The following five auxiliary commands may be issued from within any mode:

HELP	This command results in the display of a HELP page. The HELP page, entered from the top level of a GR Monitor, displays all of the operating modes available, any of which may be entered directly from that page. The HELP page entered from within an operating mode, displays instructions for changing options and specifications within that mode. Depressing the XMIT (or RETURN) keys will result in a return to the mode page from which the HELP command was issued.

RUN This command causes the mode in use to be RUN. Until the RUN command is issued, command lines can be used to change options without actually running the options selected. When RUN is added to a command line, the operating mode selected is entered, any options listed are changed, and the mode is then *run*. For example, the command line

```
#DIA TP=DSK2:NAMEA.TP;RUN
```

will cause the DIAGNOSE Mode to be entered, the TP file specification changed to read TP=DSK2:NAMEA.TP and all files required for UUT testing loaded into core for use. The command line

```
#DIA TP=DSK2:NAMEA.TP
```

on the other hand, will change the TP specification only. Note that a semi-colon (;) *must* separate the last entry from the command RUN.

EXIT or CTRL C This command causes a return to the top level of the GenRad Monitor in use from within any operating mode. CTRL C is invoked from the Teletype keyboard only. EXIT (followed by XMIT or RETURN) may be entered from either keyboard.

OS This command is used to exit to the OS/8 Keyboard Monitor from the top level of a GenRad Monitor or from within any operating mode controlled by a GenRad Monitor.

SAVE This command causes the status of the mode in use to be saved in a monitor save file. Save files are described further in Para. 3.3.6.

3.3.5 Positional Arguments

File names, devices, and options may be changed at any time by displaying the desired page on the data terminal screen, and then making the desired changes using the procedures described in Para. 3.3.4. It is also possible to change or modify sets of options through the use of positional arguments that are inserted into command lines, and issued from the GenRad Monitor top level (TEST, CAPS, or APG display).

Since the number and position of option sets may differ from mode to mode, changes to option sets located on the first page of each mode must be accomplished in an ordered sequence if they are to work correctly. For example, the first page display of the DIAGNOSE Mode contains the equate options TP, UR, W and LI (see Figure 3-5). The following command line (issued from the monitor *top level*) will change these option sets in the following manner:

```
#DIA TTY: < SYS:ABC.TP,SYS:NAMEA,DSK2:,DSK3:
     LI         TP         UR         W
```

Note that the Output specification changes the LI option, while the Input specification changes the TP, UR, and W options in the *order* listed. Table 3-4 lists the order that *must* be followed when using positional arguments to change sets of options located on the *first* page of any operating mode. Note that the table lists those options that apply to the *first* page of each mode, and that must be placed in an ordered format if the change is to be accomplished. Other options, located on the same or different pages within a mode, may also be listed in the same command string, and in *any* order. For example, the command string

```
# DIA TTY: < SYS:ABC.TP,SYS:NAMEA,DSK2:,DSK3:;PN H,R;IL=4
       LI        TP        UR          W        P     IL
```

may be issued to change the Print, and AFL IL parameter options on pages three and four of the DIAGNOSE Mode. A semi-colon (;) must separate each of the options that are not required to be listed in a specific order as defined by Table 3-4.

Any of the commands that are valid within a mode may be included within a command string that invokes that mode. Such commands must be separated by semicolons (;). For example, the command strings

```
# DIA;LI=LPT:;TP=SYS:NAMEA.TP;PY I,L

        or

# DIA;FL=2
```

are valid and will change the options specified within the DIAGNOSE Mode. Command strings may be as long as necessary, and can consist of up to 72 characters, if desired.

Commas may be inserted in positional arguments to skip any of the options listed in Table 3-4 that are *not* to be changed. For example, the PREPARE Mode order of commands is:

```
# PRE TEC:< DSK:ABC.SR,ABC.TP,SYS:USERLB,SYS:F.BC,DSK2:,DSK3:
       LI      N         T        U         LM         W
```

The N and U options can be skipped by the following command line:

```
            ┌Skip N ┌Skip U
            ↓       ↓
# PRE TEC:< ,ABC.TP,,SYS:F.BC,DSK2:,DSK3:
       LI      T        LM         W
```

Note that only one comma is required to skip the first option on the Input side (N) while two commas are necessary to skip the U option (the first comma serves as a separator between T and LM while the second comma executes the skip).

Options on the Output side of a command line may be skipped by not including them in the command string. For example, the command line

```
# PRE ,ABC.TP,,SYS:F.BC,DSK2:,DSK3:
         T        LM         W
```

changes the T, LM, and W options only.

Command strings may be terminated at any point and those sequential options that are not included will be ignored. For example, the command line

```
# PRE ,ABC.TP,,SYS:F.BC
         T        LM
```

changes the T and LM options only (the dual W option is not changed).

Table 3-4

COMMAND SEQUENCE USED TO CHANGE FIRST PAGE MODE OPTIONS

Mode	Order (or sequence) of commands
EDIT	O<I or O< or O. Where O is the Output specification and I is the Input specification. For example, # EDI $\underbrace{\text{DSK2:A.TP}}_{O}$<$\underbrace{\text{DSK3:B.TP}}_{I}$ Changes both the Input and Output specifications separately (O may differ from I). # EDI $\underbrace{\text{DSK2:A.TP}}_{O}$< Changes only the Output specification. Any Input specification is deleted. # EDI $\underbrace{\text{DSK3:A.TP}}_{O \text{ and } I}$ Changes both Input and Output specifications by setting both O and I equal to the file specified.
TRANSLATE	LI < TP,W . For example, # TRA $\underbrace{\text{LPT:}}_{LI}$ < $\underbrace{\text{SYS:A.TP}}_{TP}$,$\underbrace{\text{DSK1:,DSK2:}}_{W}$ Note that the W option is a dual specification in that it calls for two different disk surfaces.
DIAGNOSE	LI <TP,UR,W. For example, # DIA $\underbrace{\text{TTY:}}_{LI}$ < $\underbrace{\text{SYS:ABC.TP}}_{TP}$,$\underbrace{\text{SYS:USRRTN.UR}}_{UR}$,$\underbrace{\text{DSK2:,DSK3:}}_{W}$
CLEANUP	N,W. For example, # CLE $\underbrace{\text{DSK2:ABC.SR}}_{N}$,$\underbrace{\text{DSK2:,DSK3:}}_{W}$
CONVERT	LI < I,O. For example, # CON $\underbrace{\text{TEC:}}_{LI}$ < $\underbrace{\text{SYS:ABC.TP}}_{I}$,$\underbrace{\text{DSK1:ABC.CD}}_{O}$
PREPARE	LI < N,T,U,LM,W. For example, # PRE $\underbrace{\text{TEC:}}_{LI}$ < $\underbrace{\text{DSK:ABC.SR}}_{N}$,$\underbrace{\text{ABC.TP}}_{T}$,$\underbrace{\text{SYS:USERLB}}_{U}$,$\underbrace{\text{SYS:MACFIL.BC}}_{LM}$,$\underbrace{\text{DSK2:,DSK3:}}_{W}$
GENERATE	LI < N,T,U,W. For example, # GEN $\underbrace{\text{LPT:}}_{LI}$ < $\underbrace{\text{DSK2:7474.SR}}_{N}$,$\underbrace{\text{7474}}_{T}$,$\underbrace{\text{SYS:USERLB}}_{U}$,$\underbrace{\text{DSK2:,DSK3:}}_{W}$
DISPLAY	No Positional arguments may be used but options may be listed in any order via equates. For example, # DIS;$\underbrace{\text{N=SYS:ABC.SR}}_{N}$;$\underbrace{\text{DM=LIST}}_{DM}$;$\underbrace{\text{T=RANGE}}_{T}$
LIBLIST	LI < LB. For example, # LIB $\underbrace{\text{TEC:}}_{LI}$ < $\underbrace{\text{SYS:USERLB}}_{LB}$
ICDELETE	N,LB. For example, # ICD $\underbrace{\text{7400}}_{N}$,$\underbrace{\text{SYS:USERLB}}_{LB}$
ICINFO	LI < N,LB. For example, # ICI $\underbrace{\text{LPT:}}_{LI}$ < $\underbrace{\text{7474}}_{N}$,$\underbrace{\text{SYS:USERLB}}_{LB}$
LOOK	LI < I. For example, # LOO $\underbrace{\text{TEC:}}_{LI}$ < $\underbrace{\text{SYS:NAME.BC}}_{I}$
LISTALL	# LI < N,T,W. For example, # LIS $\underbrace{\text{LPT:}}_{LI}$ < $\underbrace{\text{DSK:ABC.SR}}_{N}$,$\underbrace{\text{ABC.TP}}_{T}$,$\underbrace{\text{DSK2:,DSK3:}}_{W}$

3.3.6 Monitor Save Files (SAVE Command)

Monitor save files are actually ASCII files that the user creates. The files can contain any of the specifications or options that would be required for a run in any mode controlled by a GenRad Monitor. Save files therefore provide a user with a rapid means of storing and executing the specifications and options associated with any mode.

Save files may be created by entering a mode and setting all of the specifications and options associated with that mode to the desired values. Once all options have been set, the SAVE command can be used to create and save a file that contains the states of all mode options. The SAVE command may be issued from any page and will save the states of <u>all</u> pages in the mode. As an example, the states of all DIAGNOSE Mode options can be saved by entering the DIAGNOSE Mode and establishing the desired values. The following SAVE command could then be issued:

```
#SAVE DSK:FILDIA.BC
```

When the XMIT (or RETURN) key is depressed, the file will be saved under the name specified and on the device specified. If a device is not specified, the default device (DSK:) is assumed. If a file extension is not listed, the .BC extension will be added as a default extension. The command

```
#SAVE FILDIA
```

is therefore equivalent to the SAVE command listed above.

Save files may also be created by EDIT or supplied by GR to perform specified functions. RETURN, LINE FEED, or FORM FEED characters are ignored as each file is created thus eliminating the need for extensive editing. The following listing of a typical SAVE file covering the DIAGNOSE Mode illustrates the format used:

```
TP=DSK:NAMEA.TP;     }
UR=SYS:NAMEL.UR;     }    FILE USAGE
 W=DSK0:,DSK1:;      }
LI=LPT:;
DO M,;                    DIAGNOSTIC MODES
PO D,:                    PRINT OPTIONS
IL=1;                }
IU=2;                }    AFL PARAMETERS
FL=3;                }
FU=5;                }
GO F,J;                   GUIDED PROBE OPTIONS
CK=0                 }
PR=0                 }    GUIDED PROBE PARAM
SO M,C,P,W,R,D,;          SELECT MISC OPTIONS
 M=ERRORS;
 B=100;              }
 S=200;              }
 T=0;                }
 R=PASS;             }    OPERATOR CONTROLS
RC=10;               }
SI=LOOP;             }
 C=YES;
 V=PROG;
```

Note that ON/OFF options are always listed through use of the O command (turn OFF all options, then turn ON only the options specified).

Once a save file has been created, the states listed in that file can be established from the top level of the GR Monitor by entering a command line that includes a double quote (") followed by the file name as in the following example:

```
#DIA "FILDIA.BC
```

When the XMIT (or RETURN) key is activated, the save file states will be established regardless of the states existing prior to issuing the command. If a device is not specified, the default device (DSK:) is checked first for the file listed. If the file cannot be located on DSK: the system device (SYS:) is checked next. A single space must be inserted prior to each double quote (") in the command string. If it is desired to change the contents of a save file (perhaps to make it applicable to a different device under test, or perform some other minor modifications), the GR Editor can be used to make the changes. Commands may be issued from the top level of the GR Monitor as in the following example:

```
#EDI FILDIA.BC;RUN
```

Next, depress the XMIT key to view page 1. Correct the displayed page using the keyboard edit keys. For example, delete the DIAGNOSE Mode TP= line if the save file is to apply to a series of boards with different names. Next, depress the 1C and XMIT keys to correct the page in core, and the E and XMIT keys to create a new permanent file and enter the top level of the GR Monitor.

Next, the following command line can be issued to establish the new states:

```
#DIA SYS:NAMER "FILDIA.BC
```

Since the TP= option has been deleted from the save file FILDIA.BC, the TP option will reflect the name SYS:NAMER contained in the command line. Additional boards can then be tested with the same set of options by simply changing the board name as in the following example:

```
#DIA SYS:NAMEF "FILDIA.BC;RUN
```

The only restrictions are that the file cannot contain comments and must be in the format and order shown.

Information in the save file can be temporarily overwritten by adding changes after listing the save file in the command string. For example, the command string

```
# DIA "FILDIA.BC;TP=NAMEB.TP;RUN
```

will override the save file's TP= option (NAMEA.TP) and run DIAGNOSE with TP=NAMEB.TP. Note that the contents of the save file have not been changed so that if the file is listed by itself

```
# DIA "FILDIA.BC;RUN
```

the TP= option will be listed as NAMEA.TP.

If a change is added before listing the save file in a command string, then information in the save file will take precedence over the change (the change has no effect). For example, the command string

```
# DIA SYS:NAMER "FILDIA.BC;RUN
```

is valid only if the TP= option has been deleted from the save file. If the save file TP= option had been listed as TP=NAMEB.TP, then the file NAMEB.TP would be invoked.

Save files may also include the major mode to be invoked. For example, if the first entry in the save file had been

```
DIAGNOSE;
TP=DSK:NAMEA.TP;
  .
  .
  .
```

then the command string could have been shortened to:

```
# "FILDIA.BC;RUN
```

3.3.7 BATCH Processing

A series of tasks or jobs can be performed automatically, in sequence, through use of BATCH processing with a minimum of operator intervention. For example, a user might elect to perform several different PREPARE Mode runs in sequence and then enter the DIAGNOSE Mode only if each run is error free. Any Mode controlled by a GR Monitor may be run under BATCH by creating the proper BATCH file. BATCH files are ASCII files that can be created by the EDITor, and consist of commands that would be issued at the monitor top level prefixed by two dollar signs ($$) or two pound signs (##) where:

$$	is used if it is desired to always execute the command line. If errors are detected during the run, the current job is terminated and a jump occurs to the next command line (flush the current job and continue with the next job, if any).
##	is used if it is desired to execute the command line only if the last job was successful; otherwise, jump to the next job, if any.

The EDITor can be used to establish the BATCH file from the monitor top level in the following manner:

```
#EDI BATFI.BC;RUN
```

Where BATFI.BC is the name assigned to the BATCH file. The file can be lcoated on any device. If a device is not specified, DSK: is assumed. Each BATCH file should be assigned a .BC extension. Next, the BATCH file is created by listing the appropriate command lines as in the following example:

```
$$ PRE "PREA;N=NAMEA.SR;T=NAMEA.TP
## DIA "FILDIA;TP=NAMEA.TP
$$ PRE "PREB;N=NAMEB.SR;T=NAMEB.TP
## DIA "FILDIA;TP=NAMEB.TP
$$ PRE "PREA;N=NAMEC.SR;T=NAMEC.TP
## DIA "FILDIA;TP=NAMEC.TP
```

As with save files, comments cannot be added to the file and the file must be in the format shown. A single space must also be inserted prior to each double quote ("). To enter the new BATCH file into core, depress the A and XMIT keys. Next, depress the E and XMIT keys to create a permanent file and enter the monitor top level.

BATCH jobs are executed from the monitor top level by specifying the BATCH file name in a BATCH specification. For example,

```
#BAT BATFI.BC;RUN
```

will execute the BATCH file BATFI.BC created previously by EDIT. Each command line is echoed on the Teletype as the job is performed. Errors are also listed by Teletype. If an error occurs in the interpretation or execution of any command line, the monitor flushes all batch input until the next $$ (or the end of the file) are reached and generates a

```
JOB TERMINATED;FLUSHING TO NEXT JOB
```

message. The BATCH file listed will cause the following events to occur in order:

1. The PREPARE Mode will be entered, and the save file "PREA used to establish all PREPARE Mode option states except for the file usage N= and T= options which are established by the .SR and .TP names listed inthe command line. Next, the PREPARE Mode is RUN.

2. If errors are detected during the run, the job will be flushed, and the next $$ command line executed (## command lines are only executed if the last job was successful - no errors).

3. If errors are not detected, the next job in sequence will be run (in this case ## DIA NAME.TP "FILDIA). This command line causes the DIAGNOSE Mode to be entered and the save file "FILDIA used to establish all DIAGNOSE Mode options except for the TP= option which is established by NAMEA.TP in the command line.

4. The process continues until all command lines have been executed. It should be noted that the use of the DIAGNOSE Mode is not entirely automatic because normal operator intervention is required to initiate tests, etc., however, the command line does illustrate the ease with which any Mode can be entered and executed using BATCH job files.

3.3.8 Issuing OS/8 Command From Within A GenRad Monitor

OS/8 Keyboard Monitor Commands or CCL Commands may be invoked from the top level of a GenRad Monitor (CAPS, TEST or APG) by comgining them with the OS command in a single command line. In addition, OS/8 single line commands may also be invoked from within a top level GenRad BATCH processing run through use of the OS command, if desired. Single line OS/8 commands must consist of a single command line that does not require an additional command decoder line to specify Input/Output files, etc. (the CCL commands SQUISH, SUBMIT, COPY, and DELETE fall into this category). OS/8 Keyboard Monitor commands that require an additional command line (such as R PIP, R VPIP, R EDIT, etc.) cannot be invoked through use of the OS command. Methods of issuing single line OS/8 commands from within a GenRad Monitor or from within a GenRa BATCH processing run are described further in the following paragraphs.

3.3.8.1 Issuing OS/8 Commands From A GenRad Monitor Page

OS/8 single line Keyboard Monitor commands and CCL Commands may be issued only from the top level of a GenRad Monitor through use of the OS command. If the OS command is invoked from an operating mode controlled by a GenRad Monitor (for example, from within the PREPARE Mode), an immediate return to the OS/8 Keyboard Monitor will take place without executing the OS/8 or CCL command requested. OS/8 and CCL commands are executed from the GenRad Monitor top level by specifying OS, followed by at least one space, followed by the OS/8 Keyboard Monitor or CCL command to be invoked. For example, the single command line

```
#OS SQ
```

will result in an execution of the CCL SQUISH command (SQUISH the default device DSK:) when the data terminal XMIT or Teletype RETURN keys are activated. Upon completion of the SQUISH operation, an automatic return to the GenRad Monitor top level will take place.

Whenever an OS command is issued to execute another OS/8 Keyboard Monitor or CCL command, the following events take place:

1. An OS/8 Batch file is automatically created on the System Device and assigned the name JOB.BI. The file contains the following data:

```
$JOB CAPS (TEST, or APG)
.SQ
.R CAPS (.R TEST or .R APG)
```

   Where #JOB is an OS/8 Batch command to initialize for a new job. SQ is the CCL or OS/8 Keyboard Monitor Command to be executed (in this case .SQUISH), and .R CAPS (or .R TEST or .R APG) is the command that is used to execute a return to the GenRad Monitor.

2. Next, the OS/8 Batch Mode is entered automatically and the Batch job executed (in this case, a SQUISH is performed on DSK1: and then an automatic return to the CAPS Monitor takes place). The Teletype also outputs a $JOB CAPS (or TEST or APG) message as the job is executed.

The OS/8 Batch Mode is run with the /Q (Quiet) Option invoked for each OS/8 or CCL command being executed. As a result, CCL commands that produce output (such as DATE and CORE) cannot be accessed directly from the top level of a GR Monitor. These commands may be executed from the Keyboard Monitor in the usual manner or by nesting the commands within an OS/8 Batch file and then calling OS/8 Batch through use of the CCL SUBMIT command (#OS SUB SYS:JOBA.BI).

Some examples of OS/8 or CCL commands that can be executed from a GR Monitor top level might include:

Command	Meaning
`#OS DIR =3`	List the DSK: directory on the TEC Terminal in three columns.
`#OS DIR LPT: < SYS: =4` or `#OS DIR SYS: -L =4`	List the SYS: directory in four columns by line printer.
`#OS VERI DSK1: < SYS:NAMEA/N/O`	Transfer (copy) the file NAMEA from SYS: to DSK1: and verify the transfer.

#OS SUB SYS:JOBA.BI	Call OS/8 Batch into core and execute the system file JOBA.BI, where JOBA.BI is a user furnished ASCII file that can contain a number of OS/8 Batch jobs to be executed in sequence. OS/8 Batch files must be located on the System Device.
#OS DEL SYS:JOBA.BI	Delete the System file JOBA.BI.

The following rules apply whenever OS commands are to be used to execute OS/8 Keyboard Monitor or CCL commands:

1. The OS/8 file BATCH.SV must have been saved on the System device; otherwise, a BATCH.SV NOT FOUND error message will be generated.

2. OS/8 CCL commands must be enabled; otherwise, an INPUT FAILURE error message will be generated. If the message does occur, use CTRL C to execute a return to the OS/8 Keyboard Monitor and activate CCL (R CCL) before proceeding.

3. At least one space must be inserted between the OS command and OS/8 Keyboard Monitor or CCL command to be executed; otherwise, an ERROR IN COMMAND, FLUSHING JOB message will occur.

3.3.8.2 Issuing OS/8 Commands From Within A GenRad BATCH Run

OS/8 Keyboard Monitor commands may be included as part of a GenRad BATCH processing file, if desired, and executed in sequence through use of the OS command. For example, the following GenRad BATCH file could have been prepared with the Editor and then saved on DSK1: under the name TBAT.BC:

```
$$ PRENAMEA.SR,NAMEA.TP "PREA
$$ OS SQ SYS:
$$ OS SQ
## PRE NAMEB.SR,NAMEB.TP "PREB
```

When the GenRad BATCH Mode is then run through use of the

```
# BAT TBAT.BC;RUN
```

command, both disk surfaces will be squished after completing the first PREPARE Mode run. The batch stream will then continue automatically with the next command in sequence.

All of the rules listed in Para. 3.3.8.1 also apply when issuing OS commands as part of a GenRad BATCH processing run. In addition, the following rule applies:

The OS Command cannot be inserted within another file that is to be called and executed from within the original GenRad BATCH file. The use of OS is restricted to the top level (or original) GenRad BATCH file. OS/8 Batch files that contain OS/8 Keyboard Monitor or CCL commands not prefixed by OS may be created, however, and then called through use of the SUBMIT command from within a top level GenRad BATCH file. For example, the

```
OS SUB SYS:JOBA.BI
```

command calls OS/8 Batch into core and then executes the jobs contained in the System file JOBA.BI. Care must be taken, however, to include the .R CAPS (.R APG or .R TEST) command as the last entry in the OS/8 Batch file if an automatic return to the CAPS (APG or TEST) Batch job is desired. The OS/8 command $END returns control to the OS/8 Keyboard Monitor, while executing the OS/8 file without an .R CAPS (.R APG or .R TEST) or $END command will result in the generation of an INPUT FAILURE message (use CTRL C to return to the OS/8 Keyboard Monitor).

3.3.9 GenRad REMember Program

The GenRad REMember Program (.R REM) displays the contents of the DEC TECO, DEC EDITOR AND DEC COMPILER and the DEC COMMAND REGISTERS UA, UB, and UC. The EDITOR is not supported by the CAPS IX software.

The UA, UB, and UC commands are used to remember arguments which can then be executed as CCL commands. This capability allows an operator to enter arguments which can be used to simplify repetitive CCL command entries.

If an operator must reference the same directory several times a day, it may be easier to enter the directory into register UA as

```
.UA DIR LPT: <DSK2:
```

and then simply enter the CCL Command .UA to list the contents of the directory. (The . is issued by the keyboard monitor.)

Other CCL commands can be entered into the registers in this manner, and will be executed as long as they do not require listings to the TEC Scope.

To enter the REMember program, type .R REM and the contents of TECO, EDITor, COMpiler, and Registers UA - UC will be displayed.

```
20-MAR-79

TECO : - FILE.PA

EDIT : - DSK:FILE.XX

COM  : - FILE.BN.FILE.LS <FILE.PA

UA   : - DIR LPT: <DSK2:

UB   : - LAST ENTRY

UC   : - LAST ENTRY
```

When making new entries, the display is not updated until .R REM is rerun. For further details refer to the DEC OS8 Handbook, UA, UB and UC Commands.

System Start-Up– Section 4

4.1 INTRODUCTION

Detailed system start-up procedures are contained in the standard system operating instructions supplied with each test system. The condensed start-up procedures in this section are extracted from those manuals and may be used by operators who are familiar with general test system and computer operation to start-up a test system and enter the DEC OS/8 Keyboard Monitor.

4.2 SYSTEM SOFTWARE

4.2.1 Software Required

Standard system programs and CAPS/APG programs are pre-recorded on a system cartridge which must be installed in the top drive of double disk systems (or the only drive of single disk systems) before the test system can be used. The following CAPS/APG pre-recorded system cartridges may be supplied:

Monitor Desired:

CAPS	APG
1796-9671	1796-9672

4.2.2 General Precautions

It is recommended that the following precautions be observed during all operating runs to avoid loss of data:

1. Install and remove cartridges only when the LOAD lamp is illuminated. Disk doors remain locked at all other times.

2. When a disk is rotating at operational speed (RDY Lamp illuminated), never switch to the LOAD position while data is being recorded or information could be lost.

3. If unusual noises are detected as a cartridge is brought up to operational speed (scratch sounds, etc.), return the drive to the LOAD position. When the LOAD lamp lights, remove and inspect the cartridge (it could be warped or otherwise defective).

4. To avoid loss of data, always keep cartridges away from any source of radiated emissions.

5. Always re-start the test system by following the appropriate starting procedure contained in Para. 4.3, 4.4, or 4.5.

6. Once the keyboard monitor has been entered, use the DATE command to set the system date so that subsequently created files, listings, and printouts will be correctly dated. Dates remain in effect until the test system is re-started. Many of the system programs require that a DATE be established before the program can be entered.

7. Whenever possible, back yourself up with duplicate software so that the impact of system "crashes" or other major catastrophies can be minimized (if they do occur.)

4.3 STARTING PDP8/A SYSTEMS

Use the following procedure to start-up (or re-start) PDP8/A systems.

a. Apply SYSTEM POWER.

b. Install the system cartridge in the top drive of double disk systems or the only drive of single disk systems.

c. Set the disk drive RUN/LOAD switch to RUN. Wait for the RDY lamp to light before proceeding.

d. There are three methods of bootstrapping a PDP8/A equipped system. The first is by cycling (raise then lower) the toggle switch BOOT located on the Limited Function Panel. The second is by twice depressing the pushbutton BOOT on the Programmer's console. If the OS/8 Keyboard Monitor cannot be entered by either method, proceed as follows to manually load the bootstrap:

STEP	OPERATION
1	Press MD and DISP
2	Enter 0000_8 and press LXA
3	Enter 0030_8 and press LA
4	Enter 6743_8 and press D NEXT
5	Enter 5031_8 and press D NEXT
6	Enter 0030_8 and press LA
7	Press INIT and RUN
8	When the Teletype prints a period (.) the Bootstrap operation is complete.

e. Date the files at this time. For example,

```
.DA 1-APR-79 ↓
```

dates directory files, listings, and printouts using the form DAY-MONTH-YEAR.

NOTE

> If a system DATE is not established at this time, some system modules, modes, and programs cannot be run.

f. Check that the proper SETUP Mode parameters have been established for the test system configuration in use (refer to the system Operating Instructions supplied with each test system).

4.4 STARTING PDP8/E SYSTEMS WITH DEC OR PLESSEY DISK CONTROLLERS

Use the following procedures to start-up (or re-start) PDP8/E systems equipped with DEC or Plessey controllers, and DEC or Diablo drives:

a. Apply SYSTEM POWER.

b. Install the system cartridge in the <u>top</u> drive of double disk systems or the only drive of single disk systems.

c. Observe that the WT PROT indicator lamp is off; if it is illuminated, toggle the WT PROT switch and observe that the lamp extinguishes (DEC drives only).

d. Set the disk drive RUN/LOAD switch to RUN. Wait for the RDY lamp to light before proceeding.

e. Set the computer Indicator Selector Switch to MD, and set 0000 into the SR.

f. Press the EXTD ADDR LOAD switch (Address field 0).

g. Set 0030 into the SR, and press the ADDR LOAD switch (address location 0030).

h. Enter the following INSTRuctions into the computer SR and raise the DEP switch after <u>each</u> entry:

LOCATION	INSTR
0030	6743
0031	5031

i. Set 0030 into the SR (starting address).

j. Press the ADDR LOAD, CLEAR, and CONT switches in order. A dot (.) printout should occur signifying that the keyboard monitor has been entered. Date the files at this time. For example,

```
.DA 1-APR-80 ↵
```

dates directory files, listings, and printouts using the form DAY-MONTH-YEAR.

NOTE

> If a system DATE is not established at this time, some system modules, modes, and programs cannot be run.

k. Check that the proper SETUP Mode parameters have been established for the test system configuration in use (refer to the System Operating Instructions supplied with each test system).

4.5 STARTING PDP8/E SYSTEMS WITH DSD DISK CONTROLLERS

Use the following procedures to start-up (or re-start) PDP8/E systems equipped with DSD controllers, and Diablo drives:

a. Apply SYSTEM POWER.

b. Install the system cartridge in the top drive of double disk systems or the only drive of single disk systems.

c. Set the LOAD/RUN switch to RUN. Wait for the yellow READY lamp to light.

d. Set the computer Indicator Selector Switch to MD, and set 0000 into the SR.

e. Depress the SING STEP switch. Next, press the CONT switch three times, then raise the SING STEP switch*.

f. Press the EXTD ADDR LOAD switch (address field 0).

g. Press the ADDR LOAD switch (address location 0000).

h. Enter the following INSTRuctions into the SR and raise the DEP switch after each entry:

LOCATION	INSTR
0000	0000
0001	0000
0002	7600
0003	0004
0004	6175
0005	6174
0006	5006

i. Set 0004 into the SR (starting address).

j. Press the ADDR LOAD, CLEAR, and CONT switches in order. A dot (.) printout should occur signifying that the keyboard monitor has been entered. Date the files at this time. For example,

```
.DA 1-APR-80 ↵
```

dates directory files, listings, and printouts using the form DAY-MONTH-YEAR.

NOTE

> If a system DATE is not established at this time, some system modules, modes, and programs cannot be run.

---

* Step e is required to clear a 3-cycle data break request by the disk controller

k. Check that the proper SETUP Mode parameters have been established for the test system configuration in use (refer to the System Operating Instructions supplied with each test system).

4.6 SYSTEM SHUT-DOWN

Use the following procedure to shut-down any test system (or change system cartridges) without removing power:

a. Enter the OS/8 Keyboard Monitor.

b. Enter the GR system program QUIT. This program allows a system cartridge to be removed and the same or a different cartridge inserted without having to re-start the test system again. The program should also be used prior to any equipment shut-down as it prevents quick re-starts (such as the OS/8 7600 starting address that could destroy data, if used improperly) from being accomplished. QUIT may be entered by typing the command:

```
.R QUIT ↓
```

When an up-arrow (↑) occurs, an operator may proceed.

c. Set the drive LOAD/RUN switch to LOAD. When the LOAD lamp lights, remove the cartridge.

To Re-Insert a Cartridge (Re-Start)

To re-insert a cartridge and enter the OS/8 Keyboard Monitor, proceed as follows:

a. Insert the system cartridge into the drive and set the LOAD/RUN switch to RUN.

b. When the yellow READY lamp lights, depress any Teletype key (space bar, etc.). The OS/8 Keyboard Monitor will be entered automatically.

To Remove System Power (Equipment Shut-Down)

To remove system power (once the system cartridge has been removed from the drive) set the control panel SYSTEM POWER key switch to OFF.

To re-start a powered down test system after using QUIT, proceed as follows:

a. Enter 0000 into the SR and depress the LXA switch for PDP8/A's or the EXTD ADDR LOAD switch for PDP8/E's.

b. Set 0030 into the SR, and depress the LA switch for PDP8/A's or the ADDR LOAD switch for PDP8/E's.

c. Depress the INTI and RUN switches for PDP8/A's or the CLEAR and CONT switches in order for PDP8/E's. A dot printout (.) should occur signifying that the keyboard monitor has been entered.

Prepare Mode–Section 5

5.1 INTRODUCTION

This section contains a functional description of the software used during PREPARE Mode runs. It also includes a description of all PREPARE Mode optional commands that are available for use during a simulation run, some methods of simulation using the sample network description (.SR file) and test program (.TP file) that were prepared in Section 2, methods of file management, and some special features that may be useful during a run.

5.2 PREPARE MODE SOFTWARE

5.2.1 General

Individual sections of the PREPARE Mode software can be turned ON or OFF to accomplish specific tasks during a simulation run. Each major section of software is controlled by activation of a MODE Switch on the EXECUTE SUB MODE Page (the second PREPARE Mode page displayed), and can provide the following overall features:

Sub-Mode Switch ON	Function
N	For Network Description (.SR) Analysis. Checks the basic network description (.SR file) for syntax, semantic, and other errors, and generates the data bases required by all other PREPARE sub-mode sections including AFL, the smart probe, and the DRM (FINDS) in the DIAGNOSE Mode. Also generates cross-reference data, warnings, and statistics. The major actions that take place when N is ON are described further in Para. 5.2.2.
T	For Test Program (.TP) Translation for Simulation. Expands test program macro commands (if any), and generates the translated CAPS test program data base needed by other sub-mode switches during simulation, and by the probe when the DIAGNOSE Mode is entered. The major actions that take place when T is ON are described further in Para. 5.2.3.
G	For Good Network Simulation. Simulates a good UUT. adds outputs to test program by creating a digitally auto-programmed test program (.DA file). Generates simulation data for use by other sub-modes together with AFL, the smart probe and the DRM (FINDS) in the DIAGNOSE Mode. Also provides both good board simulation results and pin activation results for test program analysis. The major actions that take place when G is ON are described further in Para. 5.2.4.

Sub-Mode Switch ON	Function
D	For Diagnosis Mode preparation. Creates the binary data bases used by the smart probe and DRM (FINDS). Also creates a test program binary file used during UUT testing in the DIAGNOSE Mode. Also causes the P sub-mode described below to be run. The major actions that take place when D is ON are described further in Para. 5.2.5.
F	For Fault Detection Simulation. Identifies and collapses detectable failures into fault classes. Simulates faulty networks and compares expected correct outputs against simulated outputs. Generates fault dictionary files for AFL use in the DIAGNOSE Mode. Also allows selective fault simulation for test program improvement. The major actions that take place when F is ON are described further in Para. 5.2.6.
Q	For Quick Lookup Preparation. This mode makes the QL data base file required in order to run the Quick Fault Dictionary Lookup (Q Option) in the DIAGNOSE Mode. The Quick Lookup may be used to quickly pass failure points to the computer-guided probe when AFL is not enabled. The major actions that take place when Q is ON are described further in Para. 5.2.8.
P	For Probe Files (.US, .UJ, .UC, & .UD) Translation. Generates the translated probe starting-point data base from a user-furnished .US file, the jumping-point data base from a user-furnished .UJ file, and the conditional connectivity database from a user-furnished .UC file. These data bases are required by the probe in the DIAGNOSE Mode only if the user wishes to specify particular points at which to start probing or points that the probe should jump to next if a particular type of failure occurs. Also generates the translated User Discrimination Data Base from a user written .UD file for use by the GenRad Pulse-catching Probe to ignore pulses. The .UD files created by using NODAL VERIFICATION are translated automatically, therefore, PREPARE mode, P option is not required for these files (refer to Appendix D.) Note that P is only required if user selected probe points are desired, (or if the pulse-catching probe is to ignore pulses) and will be entered automatically during a D sub-mode run. P is not required for normal probe operation. The major actions that take place when P is ON are described further in Para. 5.2.7.
Z	For Percent Processing Updates. Generates the CAPS data base files that require changing (because of percent processing command changes or software updates) without having to re-run the N, T, and G sub-modes of PREPARE again. Also causes the P sub-mode to be run if the D sub-mode is not enabled. The major actions that take place when **Z** is ON are described further in para 5.2.9.

Sub-Mode Switch ON	Function
I	(APG Monitor only). For incremental Test Program Preparation and Simulation. This sub-mode provides the capability of monitoring the effects of test sequences that are appended to the test program through Incremental Simulation. As the sub-mode is run, the programmer can see how each test program segment affects the simulated model through use of the PRETTY output. The E/T/C Mode described in Appendix B may also be used at this time as a test program "debugging" tool, if desired. The I sub-mode includes an interactive editor for "on-line" corrections of test program increments to be accepted or rejected as the need dictates. The major actions that take place when I is ON are described further in Appendix B.

The software activated by each sub-mode switch is described in the following paragraphs. Because the software controlled by each switch may generate CAPS or APG data base files that are required by other PREPARE Mode switches as the sub-modes are run (or needed for testing in the DIAGNOSE Mode), a sequence of operation must be established. It is not possible, for example, to run a test program translation for simulation (T ON) without first having run a network description analysis (N ON) because some of the CAPS and APG files created by N are needed by T. The operating sequences that may be used are:

Sub-Mode Switch ON	Prior Modes Needed	Sub-Mode Switch ON	Prior Modes Needed
N	None	Q	N,T,G,F
T	N	P	N,T
G	N,T	Z	N,T,G
D	N,T,G	I	N
F	N,T,G		

This restriction does not mean that a prior Sub-mode must be re-run before the Sub-mode selected can be entered (for example, that N must be re-run every time that it is desired to run T); but rather, that the prior Sub-mode must have been run previously. Once a prior Sub-mode has been run to create the necessary CAPS and APG files, it can be turned OFF, if desired. For example, G can be run after N and T have been run once. If N and T are then turned OFF, G can be run again without re-running N and T. Such a procedure assumes that (with the exception of percent processing command changes that can be re-run through the Z sub-mode without re-running the N, T, and G sub-modes) no changes have been made to the basic .SR and .TP files that would necessitate re-running N and T again. For testing purposes in the DIAGNOSE Mode, the following major mode switches must have been turned ON during a PREPARE Mode run:

Type of Diagnosis Desired	PREPARE Mode Switches ON
1. All (MANUAL, AFL, PROBE/FINDS, Quick LOOKUP) with or without user selected probing and/or Pulse Discrimination Files	N, T, G, D, F, Q

2. PROBE/FINDS or MANUAL only with or without user selected probing, and/or Pulse Discrimination Files. — N, T, G, D

The P Sub-mode may be entered atuomatically when the D or Z sub-modes are run. It may also be entered by itself so that changes to .US, .UJ and/or user written .UD files can be translated without running D again. Note that F is only required if Automatic Fault Location (AFL) or the Quick Dictionary LOOKUP are to be used in the DIAGNOSE Mode. If a Quick Dictionary LOOKUP is to be used, the Q sub-mode <u>must</u> be run <u>after</u> running F. If the MANUAL Mode only is desired (no AFL or PROBE/FINDS), it is not necessary to run the PREPARE Mode. A test program (.TP file) can be prepared and translated into a binary file for test purposes using the TRANSLATE Mode.

5.2.2 N Sub-mode (Network Description .SR Analysis)

NOTE

> Set the SELECT MISC OPTIONS Page X Option ON or OFF as appropriate, prior to running the N Sub-Mode (refer to Para. 5.3.7.13).

When the N Sub-Mode Switch is turned ON, and a PREPARE Mode run is initiated, the events shown in Figure 5-1 take place. Each of these major actions are described in the following paragraphs.

5.2.2.1 Deleting Old Files of the Same Name

Old PREPARE Mode files of the <u>same</u> name are <u>deleted</u> prior to starting every new run. Once N has been entered, it will be necessary to run <u>all</u> of the PREPARE Modes that are required for UUT testing in the DIAGNOSE Mode in order to generate the proper CAPS and APG files. This is because a change in the circuit description implies a change to <u>all</u> CAPS and APG data bases. For example, if the DIAGNOSE Mode MANUAL, AFL, and PROBE/FINDS Modes are to be used, then PREPARE Mode T, G, D, and F runs must be initiated after completing N. Note that percent processing changes to the Network Description (.SR file) do <u>not</u> require a re-run of the N, T and G sub-modes if the Z sub-mode is used.

5.2.2.2 Pre-Processor

The Pre-Processor is entered next. The pre-processor checks the basic network description (.SR file) for syntax, semantic, and other errors. Checks are performed during a series of automatic passes through the pre-processor. If errors occur, error messages are generated and execution is suppressed. If the PREPARE Mode Miscellaneous E Option has also been specified, an automatic return to EDIT will occur so that errors can be corrected "on-line". A series of options on the Miscellaneous page make it possible to print all errors on the Teletype (P ON) rather than the output device specified, and print long error messages (L ON) that define the problem in clear text rather than short error messages which are more cryptic. Pre-Processor error messages may have one of the following formats:

```
ERROR# LINE #COMMENTS

  2     7    DRIVER/SENSOR# 25 USED TWICE                 )  LONG FORMAT
  1     16   SYNTAX ERROR                                 )   (L ON)
  12    43   A CONNECTION TO 34 WAS NOT MENTIONED AT      )
             LINE #15 WHEN IC9.13 WAS FIRST CONNECTED.    )
EXECUTION SUPPRESSED_
```

```
        OR

ERROR #LINE #COMMENTS                                           SHORT FORMAT
                                                                   (L OFF)
  2      7      25
  1      16
  12     43     34 NOT MENTIONED AT LINE 15 WHEN
                IC9.13 WAS CONNECTED.
EXECUTION SUPPRESSED
```

Table 5-1 lists the complete set of pre-processor error codes that can occur. In addition, the .SR file furnished EDIT (on errors) will contain line

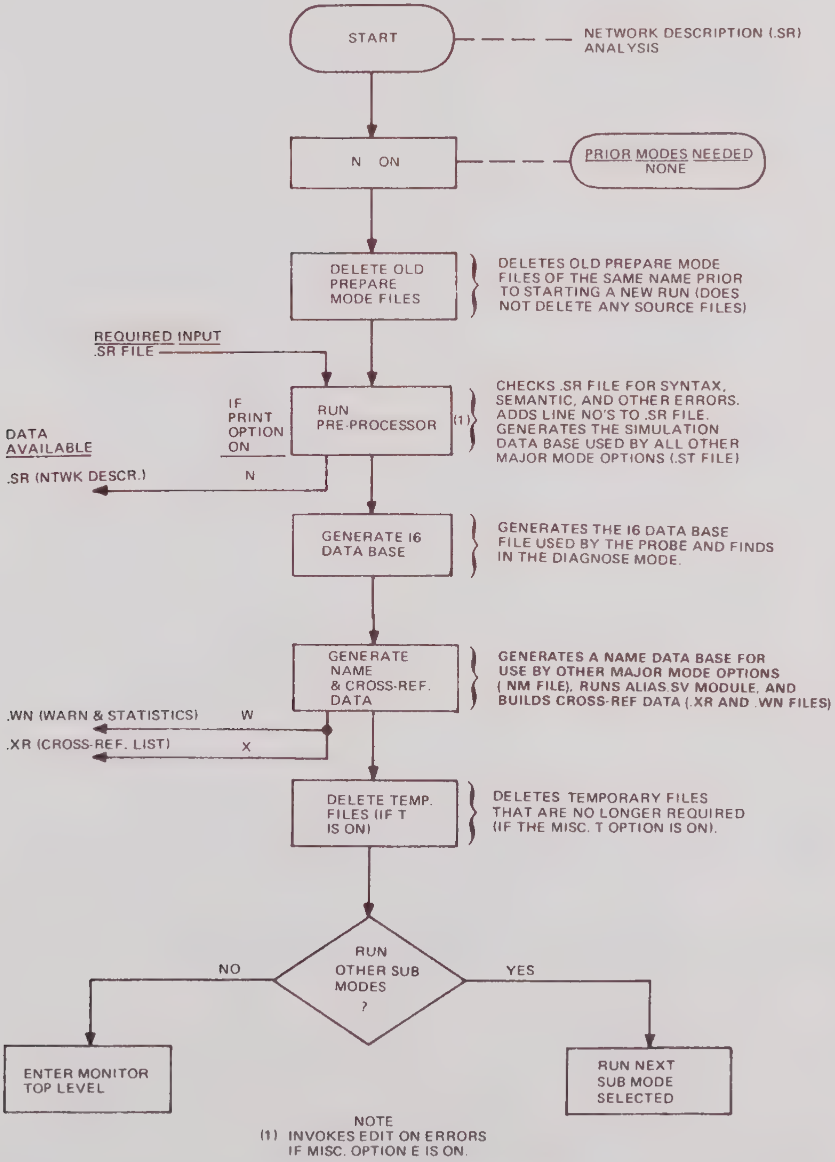

START
NETWORK DESCRIPTION (.SR) ANALYSIS
N ON
PRIOR MODES NEEDED
NONE
DELETE OLD PREPARE MODE FILES
DELETES OLD PREPARE MODE FILES OF THE SAME NAME PRIOR TO STARTING A NEW RUN (DOES NOT DELETE ANY SOURCE FILES)
REQUIRED INPUT
.SR FILE
RUN PRE-PROCESSOR
(1)
CHECKS .SR FILE FOR SYNTAX, SEMANTIC, AND OTHER ERRORS. ADDS LINE NO'S TO .SR FILE. GENERATES THE SIMULATION DATA BASE USED BY ALL OTHER MAJOR MODE OPTIONS (.ST FILE)
IF PRINT OPTION ON
DATA AVAILABLE
.SR (NTWK DESCR.)
N
GENERATE I6 DATA BASE
GENERATES THE I6 DATA BASE FILE USED BY THE PROBE AND FINDS IN THE DIAGNOSE MODE.
GENERATE NAME & CROSS-REF. DATA
GENERATES A NAME DATA BASE FOR USE BY OTHER MAJOR MODE OPTIONS (NM FILE), RUNS ALIAS.SV MODULE, AND BUILDS CROSS-REF DATA (.XR AND .WN FILES)
.WN (WARN & STATISTICS)
W
.XR (CROSS-REF. LIST)
X
DELETE TEMP. FILES (IF T IS ON)
DELETES TEMPORARY FILES THAT ARE NO LONGER REQUIRED (IF THE MISC. T OPTION IS ON).
RUN OTHER SUB MODES ?
NO
YES
ENTER MONITOR TOP LEVEL
RUN NEXT SUB MODE SELECTED
NOTE
(1) INVOKES EDIT ON ERRORS IF MISC. OPTION E IS ON.

Figure 5-1. Major Actions Initiated When The Prepare Mode N Option is ON.

Table 5-1

PRE-PROCESSOR ERROR MESSAGES

Error	Description
1	Syntax error.
2	Driver/Sensor pin mentioned twice.
3	EXTernal name defined twice.
4	Driver/Sensor pin default error. List all D/S pin numbers or none.
5	Driver/Sensor pin syntax error.
6	IC type not found in library.
7	IC defined twice (each time as a different TYPE).
8	One-shot delay specification error.
9	A pin number has been specified for a name that is not an IC.
10	A NAME has been specified that is not an IC or EXT.
11	Invalid pin number for an IC (VCC, GND, not connected, or non-existent pin).
12	Cross-referencing error.
13	Cross-referencing error.
14	No '$' data specified in Library Mode.
15	Premature end of file encountered.
16	Duplicate IC Name in Library 'NAME=' list.
17	Unknown IC feature in Library Generation Mode.
18	Unknown IC family type in Library Generation Mode.
19	Illegal threshold specification.
20	EXTernal name not number or = to D/S in Lib. Mode.
21	Not used.
22	Not used.
23	Illegal pin for load or threshold specification in Library Mode.
24	No Output feeding a chain of Inputs.
25	Cannot find a subnetwork (perhaps a library element has been deleted).
26	Unknown IC or EXTernal name.
27	Too many IC names and/or pins have been specified in additional '%' data.
28	Unknown '%' feature.
29	Invalid IC pin.
30	Not used.
31	Probe Driver/Sensor Pin conflicts with a used pin.
32	Missing SSP Code. Check CAPSX.CL.
33	Too many Outputs in a WIRED-AND node (24 maximum).
34	Unknown FAMILY Name.
35	Two or more EXTs tied together in LIB Mode at:(name).
36	No IC specified in Input File or on Monitor Page.
37	Two or more EXTINS tied together.
38	ECL and non-ECL Outputs Bussed at:(name).
39	A Library Element can't have more than 63 pins.

numbers (automatically added by CAPS), asterisks (*) and up-arrows (↑) to further define error locations. For example, error 2 at line 7 would be identified as

```
0006:  25  EXTIN  25,IC7.14 IC8.14 IC9.14 IC10.14;
0007:  26  EXTIN  25,IC7.11 IC8.11 IC9.11 IC10.11;
*                  ↑
0008:  27  EXTIN  27,IC7.5;
 .
 .
```

in the .SR file to be edited. Once the file has been corrected and is in core, invoking X and XMIT from EDIT will cause an automatic return to the Pre-Processor and the run will commence at or near the point at which it was terminated. If EDIT is exited via E and XMIT, Y and XMIT (instead of X and XMIT) a return to the top level of the Monitor will occur. The Pre-Processor automatically removes all lines with asterisks and up-arrows during the re-run. A copy of the .SR file can be obtained if the Print N option is ON. Listings are described further in Para. 5.4.

5.2.2.3 Renaming IC Devices (Alias Function)

General. During CAPS PREpare mode, it may be desirable to change IC device names so that diagnostic data are more useful. This is often true when "in-house" names or part numbers are assigned for IC names (particularly 74XXX series devices). For example, a 7400 device may be be given a part number "NJA1027" in a particular facility. If the alias name NJA1027 is listed in a CAPS circuit description (.SR) file instead of the name 7400, an error will result. A module, ALIAS.SV, allows the name NJA1027 to be associated with 7400, without having to add NJA1027 to the user's device library.

The ALIAS.SV module is run during either the N or Z sub-modes of the CAPS PREpare mode, when either the G or R option, listed on the SELECT MISCELLANEOUS OPTIONS page of PREpare mode, is turned ON. The R option renames ICs from library names to alias names, while the G option causes alias names to revert back to library names. Both options cannot be turned on at once.

ALIAS.SV modifies the .ST and .NM files (not the .SR file) so that the PRETTY file, guided-probe diagnosis, cross-reference files, and similar listing outputs contain either the alias name or the library name, as desired. The .SR file will, however, contain the library name for the device. A file listing all alias IC names, and their corresponding library names, must first be created by the user. This file is named ALIAS.LB, and must reside on DSK1:.

Operation. Operation of the ALIAS.SV module requires a properly-written CAPS circuit description (.SR) file, and an ALIAS.LB file on DSK1:. To use the ALIAS.SV module, enter the SELECT MISCELLANEOUS OPTIONS page of the CAPS PREpare mode and select either option G or R. PREpare mode is then run normally.

Contents of ALIAS.LB. The file ALIAS.LB is a cross reference file containing entries of alias names and library names. Entries in the ALIAS.LB file are made via the CAPS EDIt mode, and are of the form:

```
ALIASNAME = GENRADNAME;  /COMMENT.
```

In the ALIAS.LB file, use only one entry per line. The alias IC name (ALIASNAME) and the library name (GENRADNAME) are separated by an equal-sign (=), the entry is terminated by a semicolon (;), and any comments (optional) on a line begin with a backslash (/). The alias IC name can be up to eight alphanumeric characters long, the library name up to eight alphanumeric characters long. Spaces are not allowed as part of the name. Up to 3360 entries are normally permitted in the ALIAS.LB file. If memory is exceeded, an error message is printed, no names are changed, and ALIAS.LB is brought into the editor.

The ALIAS.LB file can contain more than one alias name for the same library name when revert mode (G option ON) is used. For example, the entries:

```
NJA1027=7400;
ABC1244=7400;
```

are permissable in revert mode, but not in rename mode (R option ON). Similarly, the ALIAS.LB file can contain more than one GenRad name for the same alias name when rename mode is used. Thus, the entries:

```
NJA1027=7400;
NJA1027=5400;
```

are permissable in rename mode, but not in revert mode.

Errors. When the ALIAS.SV module is run, the module scans the ALIAS.LB file for correct syntax. IC names are changed in the .ST and .NM files only after the scanning of ALIAS.LB is successful. If a syntax error is found, or if two (or more) library device names are found for one alias IC name, an error message will be printed, along with the line containing the error. In the event of these types of errors, no IC names are changed.

The following are examples of the error messages that can be generated by the ALIAS.SV module for faulty syntax in the ALIAS.LB file.

Error Message	Description
BAD ALIAS NAME: NJA 1027=7400;	The alias name contains a space.
BAD ALIAS NAME: NJA102799=7400;	The alias name is more than 8 characters long.
BAD ALIAS NAME: NJA:1027=7400;	The alias name contains other than alphanumeric characters.
NO ALIAS NAME: =7400;	The alias name is missing.
BAD GENRAD NAME: NJA1027=	The library name is missing.
BAD GENRAD NAME: NJA1027=740 0;	The library name contains a space.
BAD GENRAD NAME: NJA1027=74LS74000;	The library name is too long.
NO EQUAL SIGN FOUND: NJA1027 7400;	No equal sign in the entry.
NO SEMICOLON FOUND: NJA1027=7400 /COMMENT	No semicolon line terminator.
BAD COMMENT: NJA1027=7400; COMMENT	No / to begin the comment.

Messages can also occur for reasons other than faulty syntax in the ALIAS.LB file. If the warnings:

```
WARNING!  ALIAS FOUND NO NAMES TO CHANGE IN RENAME MODE.
```

or

```
WARNING! ALIAS FOUND NO NAMES TO CHANGE IN REVERT MODE.
```

are printed, the program may have been run "in the wrong direction." Check to make sure that no names should have been renamed from the .SR file. Alias IC names can be renamed by re-running N mode, or by turning ON the appropriate option (G or R) before Z mode is to be run.

If both R and G options are selected, the message:

TURN OFF EITHER R OR G

is printed when PREpare mode is run, and the SELECT MISCELLANEOUS OPTIONS page is displayed for correction.

5.2.2.4 Cross-Reference Data

After completing a good run through the pre-processor, cross-reference data is generated. The cross-reference data is then analyzed for warnings and statistics. If the Print W and X options are ON, both a cross-reference listing and a warnings & statistics listing will occur. If the X option is OFF, the .XR file will not be created during the run, however, the file may still be created later through use of LISTALL (refer to Para. 5.6.5). Warning messages in the file identify unused Inputs, Outputs, mismatches, overloads, etc. Statistical data defines the number of EXT Inputs, Outputs, busses, etc. in the .SR file and lists the amount of core needed to complete a simulation run. The cross-reference data lists board connections (as defined by the .SR file) by EXT Inputs and Outputs and by ICs. Both listings are described further in Para. 5.4.

After completing the cross-reference analysis, any temporary CAPS files that were created during the run (but are no longer required) will be deleted if the Miscellaneous T option is ON. It is thus possible to conserve disk space by deleting temporary files that will no longer be required after the run has been completed.

After completing a run, the top level of the monitor is again entered unless another sub-mode switch has been turned ON. After completing N, the T mode can be run, if desired. It is also possible for APG Monitor users to run the E/T/C Mode Equation Generation and Connectivity Analysis section or for Incremental Test Program Preparation and Simulation to be run after completing the N sub-mode of PREPARE (refer to Appendix B).

5.2.3 T sub-mode (Test Program .TP Translation)

When the T sub-mode switch is turned ON, and a PREPARE Mode RUN is initiated, the events shown in Figure 5-2 take place. Each of these major actions are described in the following paragraphs.

5.2.3.1 Macro Test Program Processor

If the Miscellaneous M Option is ON, system macro commands in the basic test program (.TP file) will be expanded by a macro processor. For example, the system macro commands:

```
.REPEAT 9;
 $ IL(7);
 $ IH(7,8);
 $ IL(7);
 $ IH(7) IL(8);
.LOOP;
```

would be expanded to include all 9 loops in the Expanded Test program (.ET file). If errors occur, execution will be suppressed and an automatic return to EDIT will take place (if the Miscellaneous E option is also ON). Macro error messages have the general format:

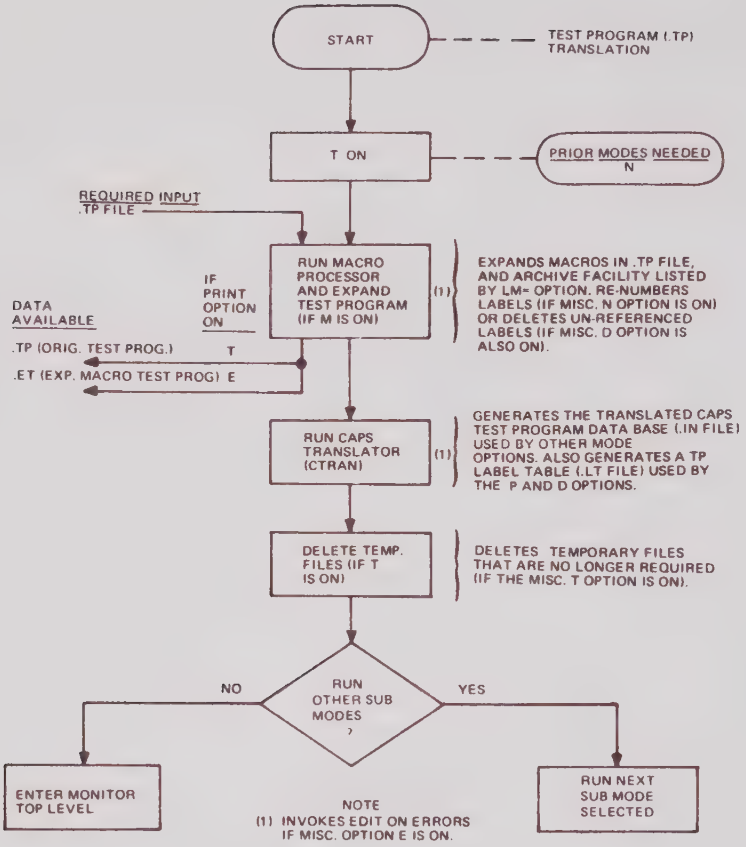

START
TEST PROGRAM (.TP) TRANSLATION
T ON
PRIOR MODES NEEDED
N
REQUIRED INPUT
.TP FILE
RUN MACRO PROCESSOR AND EXPAND TEST PROGRAM (IF M IS ON)
(1)
EXPANDS MACROS IN .TP FILE, AND ARCHIVE FACILITY LISTED BY LM= OPTION. RE-NUMBERS LABELS (IF MISC. N OPTION IS ON) OR DELETES UN-REFERENCED LABELS (IF MISC. D OPTION IS ALSO ON).
IF PRINT OPTION ON
DATA AVAILABLE
.TP (ORIG. TEST PROG.) T
.ET (EXP. MACRO TEST PROG) E
RUN CAPS TRANSLATOR (CTRAN)
(1)
GENERATES THE TRANSLATED CAPS TEST PROGRAM DATA BASE (.IN FILE) USED BY OTHER MODE OPTIONS. ALSO GENERATES A TP LABEL TABLE (.LT FILE) USED BY THE P AND D OPTIONS.
DELETE TEMP. FILES (IF T IS ON)
DELETES TEMPORARY FILES THAT ARE NO LONGER REQUIRED (IF THE MISC. T OPTION IS ON).
RUN OTHER SUB MODES ?
NO
YES
ENTER MONITOR TOP LEVEL
RUN NEXT SUB MODE SELECTED
NOTE
(1) INVOKES EDIT ON ERRORS IF MISC. OPTION E IS ON.

Figure 5-2. Major Actions Initiated When The Prepare Mode T Option is ON.

```
ERROR DETECTED BY MACRO PROCESSOR AT TEST NO:

******
0+4     SYNTAX ERROR
0+5     SYNTAX ERROR
EXECUTION SUPPRESSED
```

Table 5-2 lists some possible macro processor error messages that can occur. As with the pre-processor, depressing the X and XMIT keys (once the corrected test program is in core) will result in an immediate return to the macro processor. It is thus possible to continue to return to EDIT and correct errors "on-line" until an error free .ET file is available. If the Miscellaneous N option is ON, each labelled command in the basic test program will be re-numbered in the expanded test program (.ET file). The re-numbering sequence starts with 1 and goes up in unbroken sequence (1,2,3,4, etc.). If the Miscellaneous D option is also ON (with N) unreferenced labels (labels not referenced by CALLS, etc) will be deleted from the expanded test program, thus tending to speed-up program execution time in the DIAGNOSE Mode.

User-styled Macros that are defined through use of a Test Program

```
.DEFINE'NAME';
```

Table 5-2

TYPICAL TEST PROGRAM MACRO PROCESSOR ERROR MESSAGES

Message Format

```
'MACRO' HAS FOUND NO ROOM ON WORKING DEVICE '0'
SORRY, 'MACRO' START-UP PROCEDURE WAS INCORRECT.
ERROR DETECTED BY MACRO PROCESSORAT TEST NO:
**WARNING** UNTIL CONDITION NOT MET
SYNTAX ERROR
FLOATING CONVERSION ERROR
MULTIPLE DEFINITION OF MACRO NAME
ERROR REPORT FILE FULL (WORKING DEVICE '0')
MULTIPLE STATEMENT NUMBERS ON SAME STATEMENT
STATEMENT NUMBER GREATER THAN 8,388,607
TOO MANY STATEMENT NUMBERS
MULTIPLE USE OF SAME STATEMENT NUMBER
INSUFFICIENT SPACE ON WORKING DEVICE '0'
READ PAST ^Z (NO 'END;' STATEMENT)
'REPEAT' ACTIVE AT 'END' STATEMENT
'CASE' ACTIVE AT 'END' STATEMENT
MACRO ACTIVE AT 'END' STATEMENT
INSUFFICIENT SPACE ON WORKING DEVICE '1'
BRANCH NUMBER GREATER THAN 8,388,607
UNDEFINED STATEMENT NUMBER
STATEMENT NUMBER ON DEFINE STATEMENT
'NESTED' MACRO DEFINITION
SYNTAX ERROR IN DEFINE STATEMENT
TOO MANY DUMMY ARGUMENTS
ATTEMPT TO REDEFINE DUMMY ARG IDENTIFIER
IDENTIFIER SPECIFIES NONEXISTANT DUMMY ARG
TOO MANY DUMMY ARG IDENTIFIERS
TOO MANY MACRO DEFINITIONS
CALL TO UNDEFINED MACRO
NON-EXISTANT DUMMY ARG
REPEAT ACTIVE AT 'ENDM'
RECURSIVE MACRO CALLS ARE ILLEGAL
READ PAST ^Z (NO 'ENDM;' STATEMENT)
UNDEFINED ARGUMENT
'REPEAT' COUNT MISSING OR ZERO
'REPEAT/LOOP' OVERLAPS PROGRAM LEVELS
'CASE' BLOCK OVERLAPS PROGRAM LEVELS
READ PAST ^Z IN ARGUMENT
MULTIPLE DEFINITION OF PIN NAME
TOO MANY REPEATS AND/OR MACROS ACTIVE
DUPLICATE OR EXCLUSIVE PARAMETER
INCLUSIVE PIN SPECIFICATION BACKWARDS
PIN NOT IN SYSTEM
LOGIC PIN SPECIFIED UNDEFINED OR WRONG CONTEXT PINS:
QUESTION MARK (?) FOUND IN TEXT ARGUMENT
UNDEFINED PIN NAME
INCLUSIVE PINS ARE EQUAL
TOO MANY PINS IN PATTERN
NO PIN LIST FOR PATTERN
ILLEGAL ARGUMENT INSERTION, NO MACRO ACTIVE
MORE THAN 4095 STATEMENTS SINCE STATEMENT #
READ PAST ^Z IN 'DEFINE' STATEMENT
REPEAT-STOPLOOP FAILED TO PRODUCE UNTIL'S COND
CASE-STOPCASE FAILED TO PRODUCE UNTIL'S COND
MACRO HAD NO ROOM TO LOAD ARCHIVE HANDLER
USR FAILED WHEN LOADING THE ARCHIVE HANDLER
USR COULD NOT FIND THE ARCHIVE HANDLER IN CORE
MACRO NAME DOES NOT MATCH .DEFIN IN LIBRARY
SEMI COLON(;) FOUND IN TEXT ARGUMENT
OCTAL NUMBER CONVERSION ERROR
CAPS-RECORD-FAST-RTI PROGRAM BLOCKS OVERLAP
```

command, and then entered into the body of a test program using the

.'NAME';

command will be expanded as the T sub-mode is run. If the User-styled Macro is included in the body of a test program (using .'NAME'), but is not defined in the test program itself (that is, a .DEFINE 'NAME' command is not included in the test program), the macro may be inserted automatically if any of the following conditions exist:

1. If the macro is defined by a .DEFINE 'NAME' command that is included as part of an Archive File, and the device and file are specified by the PREPARE Mode FILE USAGE Page LM= Option, then the Macro will be inserted into the test program automatically.

2. If the Macro defined by the .DEFINE 'NAME' command is not included in the ARCHIVE Facility specified by the LM= Option, an automatic search of the test system Macro Library (DSK1:MACLIB.AR) will take place, and the Macro will be inserted into the test program, if located; otherwise, an UNDEFINED MACRO Message will be generated.

Methods of inserting files into the ARCHIVE Facility are described in Para. 3.2.6. Test program Macro Commands are described further in the Programming Instructions.

CAPS also creates a .PN file that includes any alphanumeric names used in the network description (.SR file). The .PN file is merged into the .ET file at this time (whether M is ON or OFF) so that any alphanumeric names used in the .SR file will apply (regardless of whether ASSIGN PNAME commands have been included in the .TP file by the user to cover the alphanumeric names used in the .SR file or not). If the .SR file does <u>not</u> contain any alphanumeric names, the following message will occur just prior to the .ET file listing:

```
NO ALPHANUMERIC PIN NAMES FOUND IN (NAME.SR)
```

This message is actually the first part of the .ET file. If the .SR file <u>does</u> contain alphanumeric names, the following message will occur just prior to the .ET file listing:

```
PIN NAMES FROM (NAME.SR)
ASSIGN PNAME A6=76;
```

If the .TP file has <u>not</u> been changed to reflect the alphanumeric names used in the .SR file, ASSIGN PNAME statements can be added at this time. Note that it is <u>not</u> necessary to add ASSIGN PNAME statements to the .TP file as the .PN file will change the D/S pin symbol table so that both the .SR and .TP file name assignments will apply. However, it might be desired to change the .TP file at this time so that the .TP and .ET file listings will reflect the .SR file PNAME assignments.

5.2.3.2 <u>CAPS Translator (CTRAN)</u>

The basic test program or expanded test program (.ET file) is run through a CAPS Translator (CTRAN) next. The translator checks the test program for syntax and other errors at this time. It also builds a test program data base and label table that are used by other CAPS sub-modes during the simulation run, and by the probe/clip when DIAGNOSE Mode is entered. As with the preprocessor and test program macro processor, it is possible to return automatically to EDIT on errors if the Miscellaneous E option is ON. Error messages have the general format:

```
ERRORS DETECTED BY CTRAN:

E14 AT 20   ON D/S PIN 5
E14 AT 20   ON D/S PIN 12
EXECUTION SUPPRESSED
```

Table 5-3 defines the translator error codes that can occur. Once a corrected test program is in core, depressing X and XMIT from EDIT will invoke an automatic return to the CAPS Translator (or macro processor, if the Miscellaneous M option is ON).

After completing a successful translation, any temporary files that were created during the run (but are no longer required) will be deleted if the Miscellaneous T Option is ON.

Table 5-3

TRANSLATOR ERRORS

```
/ BLOCK MESSAGES. WHEN TRANS.SV AUTOMATICALLY INSERTS A BLOCK
/       STATEMENT IN THE BINARY FILE (.T?) IT PRINTS A MESSAGE
/       OF THE FORM "B1 AT 1100+1", "B2 AT 2230+1", AND SO FORTH

/ WARNING MESSAGES. IN SOME STATEMENTS, PIN NAMES (PNAMES) CONFLICT
/       WITH SYSTEM KEYWORDS    IN THESE CASES, THE KEYWORD IS USED
/       INSTEAD OF THE PIN NAME, AND A WARNING IS ISSUED AS EACH
/       PNAME IS ASSIGNED   THESE WARNINGS INDICATE BOTH THE INVALID
/       PIN NAMES AND THE VALID KEYWORDS FOR SPECIFIC STATEMENTS
/               IF A TEXT STRING BEGINS WITH THE ABBREVIATED FORM
/       OF A KEYWORD, THE ENTIRE KEYWORD IS ASSUMED.  FOR EXAMPLE
/       CON, CONA, AND CONXXX, AT THE BEGINNING OF A TEXT STRING,
/       MEAN CONTINUE (SEE WARNING 1)

/ W1    ASSIGNED PNAME CONFLICTS WITH EXEC STATEMENT KEYWORD ACV,
/       AR, CON(TINUE), COU(NT), DCI, DCV, DCVIF, DCVP(RB), FAIL
/       FRA(TIO), FREQ, OC(TAL), PSI, PSV, RMU, TIME, TMU

/ W2    ASSIGNED PNAME CONFLICTS WITH SCANNER SPECIFICATION KEYWORD
/       A, C, GSENSE, GUARD, H, HI, IM, IP, LO, MA(SCAN),
/       MB(SCAN), RELPRO, SA(SCAN), SB(SCAN), SH(ELF), VM, VP

/ BASIC TRANSLATOR DEFINED ERRORS

/ E1    SYNTAX ERROR
/ E2    FLOATING POINT CONVERSION ERROR
/ E3    READ PAST END OF FILE
/ E4    DUPLICATE PARAMETER OR EXCLUSIVE PARAMETER
/ E5    REQUIRED PARAMETER OMITTED
/ E6    INCLUSIVE PIN SPECIFICATION BACKWARDS
/ E7    PIN NOT IN SYSTEM
/ E8    MULTIPLE COMMAND NUMBERS ON SAME STATEMENT
/ E9    NOT ENOUGH ROOM ON VIRTUAL (1ST WORKING) DEVICE
/ E10   INTEGER PARAMETER ZERO OR TOO LARGE
/ E11   DELAY SPECIFICATION CONVERSION ERROR
/ E12   DUPLICATE LABEL
/ E13   UNDEFINED PIN NAME
/ E14   LOGIC PIN SPECIFIED IS UNDEFINED OR IN WRONG CONTEXT
/ E15   TOO MANY LABELS (4095 MAX)
/ E16   BLOCK TOO LARGE
/ E17   UNDEFINED LABEL
/ E18   MACRO STATEMENT ENCOUNTERED
/ E19   COUNT PARAMETER TOO LARGE OR SMALL
/ E20-E27        <NO CURRENT MEANING>
/ E28   OCTAL NUMBER CONVERSION ERROR
/ E29   MULTIPLE DEFINITION OF PIN NAME
/ E30   TOO MANY PIN NAME DEFINITIONS
/ E31   ITT OR IST TOO LARGE
/ E32   INVALID EXEC ADDRESS
/ E33   4096 STATEMENTS WITHOUT A LABEL
/ E34   1796 SCANNER TERMINAL LIST IS INVALID
/ E35   1796 SCANNER CHANNELS MISUSED (TOO MANY, REUSED, CONFLICTING)
/ E36   1796 SCANNER DEFAULTS AND EXPLICIT SPECS MIXED
/ E37   ANALOG PROBE CONNECTION (TERMINALS OR CHANNEL) NOT GIVEN
/ E38   DIRECT SHELF CONNECTION NOT POSSIBLE AT THIS TERMINAL
/ E39   MORE THAN 4095 PATTERNS INSIDE A FAST BLOCK
/ E40   FAST DO LOOP ERROR. NESTED,TOO MANY, TEST LGC INSIDE
/ E41   NON-FAST PINS USED IN FAST BLOCK
/ E42   NEED SET/TEST LGC BEFORE THIS STATEMENT




/ E43   CAPS/RECORD/FAST CONFLICT WITH RTI
/ E44   RTI LOGIC STATEMENT EXCEEDS OPERATION/PIN LIMIT
/ E45   CGEN STATEMENT EXCEEDS 32 EVENTS
/ E46   CGEN DELAYS OUT OF RANGE OR ORDER
/ E47   CGEN DELAYS MUST BE 25NSEC INCREMENTS

/ CAPS TRANSLATOR DEFINED ERRORS

/ E50   ATTEMPTED BRANCH IN CAPS MODE
/ E51   TOO MANY CAPS SECTIONS (4095 MAX)
/ E52   MISUSE OF CAPS 'NAME'
/ E53   TOO MANY STEPS IN CAPS SECTION
/ E54   BAD CAPS DEFINED PIN SPECIFICATION
/ E55   NO CAPS SECTION OF CORRECT NAME
/ E56   INITIALIZATION MISUSED
/ E57   SET PULSE ERROR (DOES NOT MATCH CAPS PIN DEFINITIONS)
/ E58   EXCESSIVE DO NESTING (9 LEVELS MAX)
/ E59   UNIGNORED NON-CAPS DEFINED PIN WITHIN A RECORDED CAPS SECTION
/ E60   NEED ASSIGN LOGIC INSIDE CAPS SECTION BEFORE THIS STATEMENT
/ E61   PINS NOT PROPERLY ASSIGNED
/ E62   PINS SPECIFIED HAVE ALREADY BEEN DEFINED
/ E63   MACRO STATEMENTS ARE NOT ALLOWED
/ E64   WARNING. CAPS EXTOUT DEFINED AS DRIVER
/ E65   WARNING: CAPS EXTIN DEFINED AS SENSOR
/ E66   BAD DO LOOP COUNT
/ E67   WARNING: AN UNMATCHED CAPS SECTION WAS IGNORED
/ E68   WARNING. CAPS EXTERNAL NOT ASSIGNED
/ E69   TEST LGC FOUND INSIDE DO LOOP IN RECORDING SECTION
/ E70   CAPS PROBE PIN MISUSED IN TEST PROGRAM
/ E71   FAST BLOCK RESTRICTION VIOLATED
/ E72   FAST BLOCK OVERLAPS CAPS SECTION
/ E73   ONLY FAST PINS MAY CHANGE WITHIN A FAST BLOCK
```

After completing a T sub-mode run, the top level of the Monitor is again entered unless another sub-mode switch has been turned ON. After completing T, the G and P sub-modes can be run, if desired. If both G and P are turned ON, the G Mode will be entered next.

5.2.4 G sub-mode (Good Network Simulation)

When the G Sub-Mode switch is turned ON and a PREPARE Mode RUN is initiated, the following major actions take place (see Figure 5-3):

1. A model of the good UUT is simulated by the CAPS software using inputs from the N and T runs just completed. During the good board simulation, Output patterns are generated by the simulator. The outputs are then translated by an Analysis of Good Network (AGN) translator into good results, good value data bases, and AFD/AFL comparison files that are used by the AFD Module during a fault simulation run (F ON) and by the AFL Module and PROBE/FINDS during logic device testing in the DIAGNOSE Mode.

2. Good Outputs are then merged with the expanded test program to create a digitally autoprogrammed test program (.DA file) . The .DA file is a complete test program that includes both Input and Output patterns. If Output patterns were included in the original test program (.TP file), they will be checked against the simulated Output patterns in the .DA file with any differences included in a difference report (.DF file). If the Print V option is ON, a listing of the difference report will occur. If the Print A option is ON, a listing of the .DA file will also occur. Both listings are described further in Para. 5.4.

3. If the Print G option is ON, Good Board Simulation Results will be compiled and output to the device specified. Good Board Simulation Results list the states of all Inputs and Outputs grouped by EXT pin and by Device pin for the good board simulation. Test steps in the listing correspond to numbers included in comments that have been added to the .DA file at every ten steps. Good board simulation results are defined further in Para. 5.4.

4. If the Print P option is ON, a pin activation results listing will be compiled and output to the device specified. This listing defines the first and last times that each Input, Output, and Device pin is driven low and high, the number of transitions that occur, etc. Full details are provided in Para. 5.4.

5. After completing a successful G Sub-Mode run, temporary files that were created during the run (but are no longer required) will be deleted if the Miscellaneous T option is ON.

6. After completing a G Sub-Mode run the top level of the Monitor is again entered unless another sub-mode switch has been turned ON. After completing G, the D, F, and P sub-modes can be run.

It is also possible for APG Monitor users to run Activity Tracking in the E/T/C Mode after completing an N,T, and G run, if desired (refer to Appendix B). CAPS or APG Monitor users may also use the Z sub-mode of PREPARE for percent processing updates after the N,T, and G sub-modes have been run.

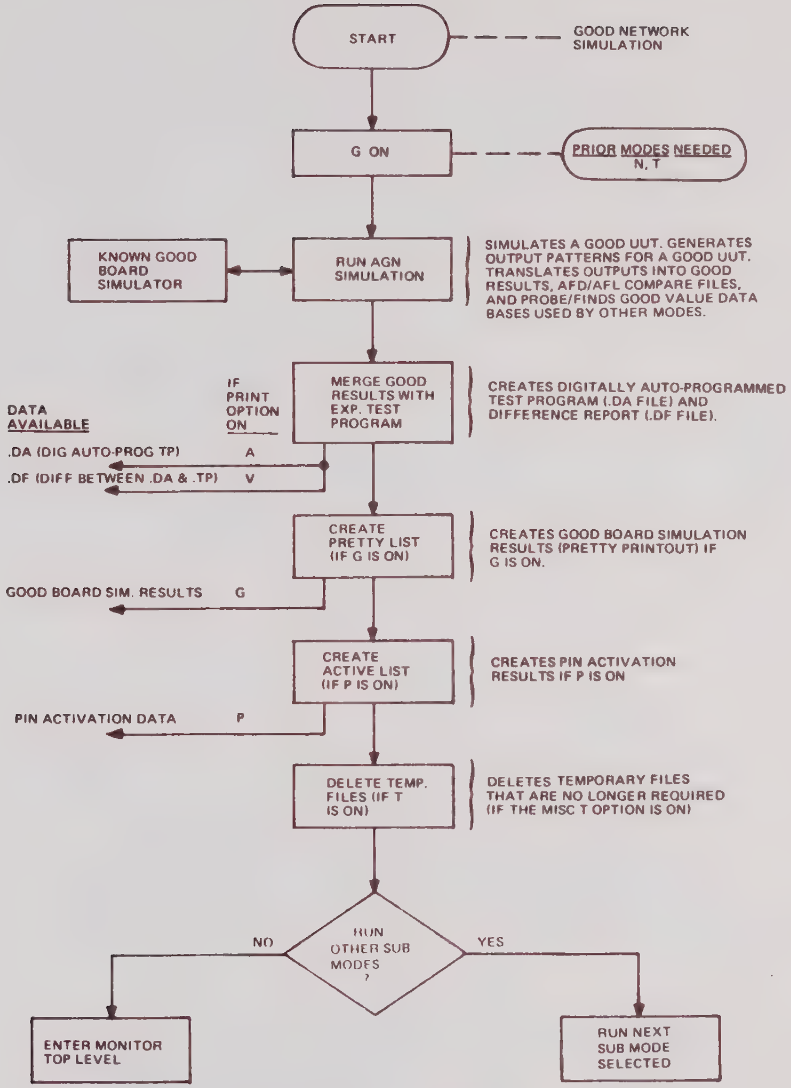

START
GOOD NETWORK SIMULATION
G ON
PRIOR MODES NEEDED N, T
KNOWN GOOD BOARD SIMULATOR
RUN AGN SIMULATION
SIMULATES A GOOD UUT. GENERATES OUTPUT PATTERNS FOR A GOOD UUT. TRANSLATES OUTPUTS INTO GOOD RESULTS, AFD/AFL COMPARE FILES, AND PROBE/FINDS GOOD VALUE DATA BASES USED BY OTHER MODES.
MERGE GOOD RESULTS WITH EXP. TEST PROGRAM
CREATES DIGITALLY AUTO-PROGRAMMED TEST PROGRAM (.DA FILE) AND DIFFERENCE REPORT (.DF FILE).
DATA AVAILABLE
IF PRINT OPTION ON
.DA (DIG AUTO-PROG TP) A
.DF (DIFF BETWEEN .DA & .TP) V
CREATE PRETTY LIST (IF G IS ON)
CREATES GOOD BOARD SIMULATION RESULTS (PRETTY PRINTOUT) IF G IS ON.
GOOD BOARD SIM. RESULTS G
CREATE ACTIVE LIST (IF P IS ON)
CREATES PIN ACTIVATION RESULTS IF P IS ON
PIN ACTIVATION DATA P
DELETE TEMP. FILES (IF T IS ON)
DELETES TEMPORARY FILES THAT ARE NO LONGER REQUIRED (IF THE MISC T OPTION IS ON)
RUN OTHER SUB MODES ?
NO
YES
ENTER MONITOR TOP LEVEL
RUN NEXT SUB MODE SELECTED

Figure 5-3. Major Actions Initiated When The Prepare Mode G Option is ON.

5.2.5 D Sub-Mode (Diagnosis Mode Preparation)

When the D Sub-Mode Switch is turned ON and a PREPARE Mode RUN is initiated, the following major actions take place (see Figure 5-4):

1. Binary data bases required by the PROBE and FINDS in the DIAGNOSE Mode are prepared. If the I sub-mode has been run previously (instead of G), then the .DA file will be created when the D sub-mode is entered.

2. The digitally autoprogrammed test program (.DA file) is next translated to obtain the test binary file required for UUT testing in the DIAGNOSE Mode. Translated test binary files are assigned file extensions based on how the SETUP Mode SYSTEM CONFIGURATION ST= option has been set. One of the following file extensions may be generated at this time:

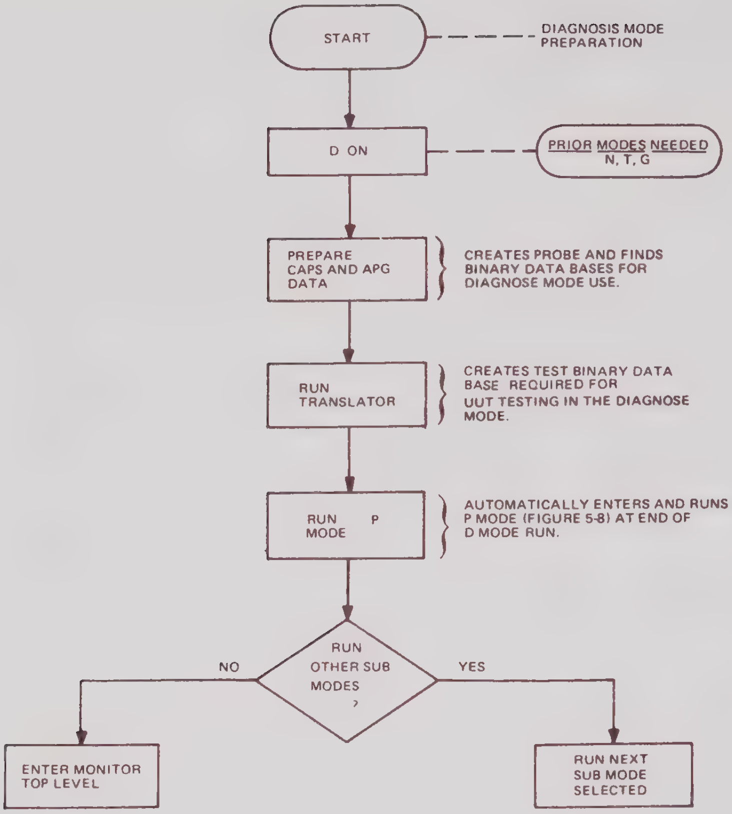

START
DIAGNOSIS MODE PREPARATION
D ON
PRIOR MODES NEEDED N, T, G
PREPARE CAPS AND APG DATA
CREATES PROBE AND FINDS BINARY DATA BASES FOR DIAGNOSE MODE USE.
RUN TRANSLATOR
CREATES TEST BINARY DATA BASE REQUIRED FOR UUT TESTING IN THE DIAGNOSE MODE.
RUN P MODE
AUTOMATICALLY ENTERS AND RUNS P MODE (FIGURE 5-8) AT END OF D MODE RUN.
RUN OTHER SUB MODES ?
NO
YES
ENTER MONITOR TOP LEVEL
RUN NEXT SUB MODE SELECTED

Figure 5-4. Major Actions Initiated When The Prepare Mode D Option is ON.

SETUP Mode ST=option	File Extension Generated
ST=A or B	.TA (For 1792A/B Test Systems)
ST=C or D	.TD (For 1792C/D Test Systems)
ST=FIVE	.TL (For 1795 Test Systems)
ST=SIX	.TV (For 1796 Test Systems)
ST=NINE	.TL (For 1799 Test Systems)

3. Next, the P sub-mode is entered automatically and any user-furnished .US, .UJ, .UC, and/or user-written .UD files are translated. When D is specified, P will be run whether it is enabled or not. If .US, .UJ, .UC and/or .UD files are available, they will be translated at this time (refer to para 5.2.7).

4. After completing the D and P sub-mode runs, the top level of the Monitor is again entered unless another sub-mode switch has been turned ON. After completing D, and P, F may be run if enabled.

5.2.6 F sub-mode (Fault Detection Simulation)

When the F Sub-Mode Switch is turned ON and a PREPARE Mode RUN is initiated, the events shown in Figure 5-5 take place. Each of these major actions are described in the following paragraphs. If the I sub-mode has been run previously (instead of G) then the AFD/AFL Compare File (.CM File) normally created during a G sub-mode run, will be created as the F sub-mode is entered. This file is needed prior to starting an AFD simulation run.

5.2.6.1 AFD Simulation

The Automatic Fault Detection (AFD) Module identifies and collapses detectable failures into fault classes. Faulty networks are then simulated. Test program outputs, obtained from a good board simulation in the G Mode, are then compared against the simulated faulty outputs, to evaluate the efficiency of the test program (how good is it at detecting modelled faults?). It also provides a set of diagnostic (fault dictionary) files that are used by AFL for fault analysis

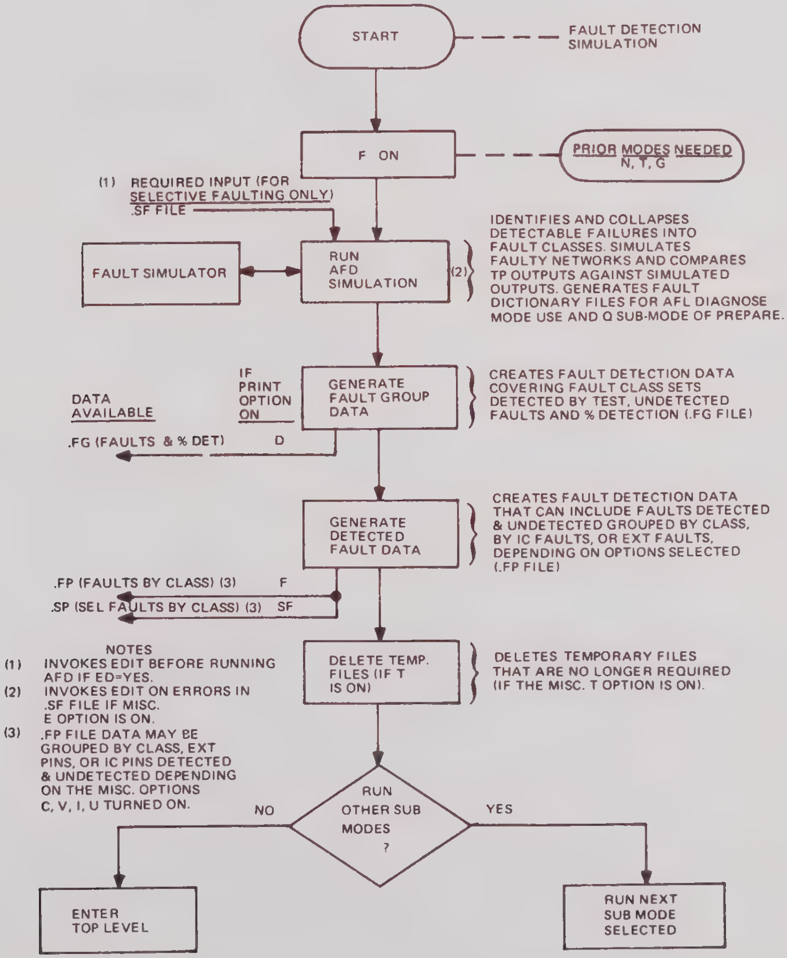

START
FAULT DETECTION SIMULATION
F ON
PRIOR MODES NEEDED
N, T, G
(1) REQUIRED INPUT (FOR SELECTIVE FAULTING ONLY)
.SF FILE
FAULT SIMULATOR
RUN AFD SIMULATION
(2)
IDENTIFIES AND COLLAPSES DETECTABLE FAILURES INTO FAULT CLASSES. SIMULATES FAULTY NETWORKS AND COMPARES TP OUTPUTS AGAINST SIMULATED OUTPUTS. GENERATES FAULT DICTIONARY FILES FOR AFL DIAGNOSE MODE USE AND Q SUB-MODE OF PREPARE.
DATA AVAILABLE
IF PRINT OPTION ON
GENERATE FAULT GROUP DATA
CREATES FAULT DETECTION DATA COVERING FAULT CLASS SETS DETECTED BY TEST, UNDETECTED FAULTS AND % DETECTION (.FG FILE)
.FG (FAULTS & % DET)
D
GENERATE DETECTED FAULT DATA
CREATES FAULT DETECTION DATA THAT CAN INCLUDE FAULTS DETECTED & UNDETECTED GROUPED BY CLASS, BY IC FAULTS, OR EXT FAULTS, DEPENDING ON OPTIONS SELECTED (.FP FILE)
.FP (FAULTS BY CLASS) (3)
F
.SP (SEL FAULTS BY CLASS) (3)
SF
NOTES
(1) INVOKES EDIT BEFORE RUNNING AFD IF ED=YES.
(2) INVOKES EDIT ON ERRORS IN .SF FILE IF MISC. E OPTION IS ON.
(3) .FP FILE DATA MAY BE GROUPED BY CLASS, EXT PINS, OR IC PINS DETECTED & UNDETECTED DEPENDING ON THE MISC. OPTIONS C, V, I, U TURNED ON.
DELETE TEMP. FILES (IF T IS ON)
DELETES TEMPORARY FILES THAT ARE NO LONGER REQUIRED (IF THE MISC. T OPTION IS ON).
RUN OTHER SUB MODES ?
NO
YES
ENTER TOP LEVEL
RUN NEXT SUB MODE SELECTED

Figure 5-5. Major Actions Initiated When The Prepare Mode F Option is ON.

in the DIAGNOSE Mode, and by the Q sub-mode of PREPARE. If the Print D option is ON, a listing of the Fault Detection Data (.FG file) organized by test number, fault class sets detected, statistics, % detected, and undetected faults can be obtained. If the Print F option is ON, fault detection data organized by class, EXT pins, or IC pins detected and undetected can be obtained (.FP file). The .FP file content is determined by which of the Miscellaneous C, V, I, or U options are turned ON and is described further in Para. 5.4. Once a fault simulation run has been completed (or at least started), selective faults may also be simulated through the execution of commands contained in a Selective Fault (.SF) file. Details are contained in Para. 5.2.6.4.

When a successful fault simulation run has been completed, any temporary files that were created during the run (but are no longer required) will be deleted if the Miscellaneous T option is ON.

After completing an F sub-mode run the top level of the Monitor is entered unless another sub-mode switch has been turned ON. After completing F, the P and Q Sub-Modes can be run.

The following paragraphs further define and explain the fault simulation process together with any abbreviations used. In addition, a detailed description of Selective Faulting is also included. The new reader may wish to skip over this material, returning to it at a later time. The description of the Probe Files Translation (P ON) contained in Para. 5.2.7 may also be classified as optional reading as it is only required if user selected probe points or points to be ignored by the pulse-catching probe are to be specified. The new reader may therefore wish to proceed to the description of PREPARE Mode options (Para. 5.3) at this time.

5.2.6.2 Fault Simulation

The following definitions may be of value in understanding how faults are simulated when F is ON:

1. Abbreviations Used. The following abbreviations may be used:

   Stuck-at-one is abbreviated as SA1

   Stuck-at-zero is abbreviated as SA0

   IC9 pin 2 SA1 is abbreviated as IC9.2-1

   IC9 pin 2 SA0 is abbreviated as IC9.2-0

2. Fault Class and Equivalent Faults. A set of modeled faults that, as observed from the board Output pins, are equivalent in their behavior no matter how many Input patterns are applied to the board. In the simple circuit shown in Figure 5-6 for example,

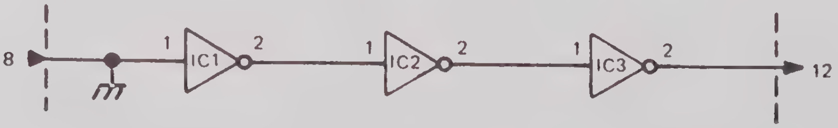


Figure 5-6. Fault Classes and Equivalent Faults.

it would be impossible to distinguish between IC1.1-0 and IC2.2-0 or IC2.1-1 and IC3.2-1, etc. no matter how many Input patterns are applied to the circuit. CAPS and APG therefore list all of the faults in the same fault class.

3. Fault Collapsing. The process of identifying and collapsing each modeled fault into fault classes and equivalent faults.

4. Fault Dictionary. A data base that contains information on the behavior of modeled faults. The test step at which a fault is detected and external values at the failing test step are recorded. The AFL Module uses these values during device testing in the DIAGNOSE Mode.

5. Fault Group. A set of fault classes that have identical behavior up to and including the failing test. As an example, if the circuit shown in Figure 5-7 had a 0110 Input pattern applied at test step 10, the expected Output response would be a 100 pattern if no errors existed. When the faults IC2.2-0, and IC3.2-0 (or IC5.3-1) are simulated, the same external signature (110) will be generated. All of the faults are therefore listed in the same fault group.

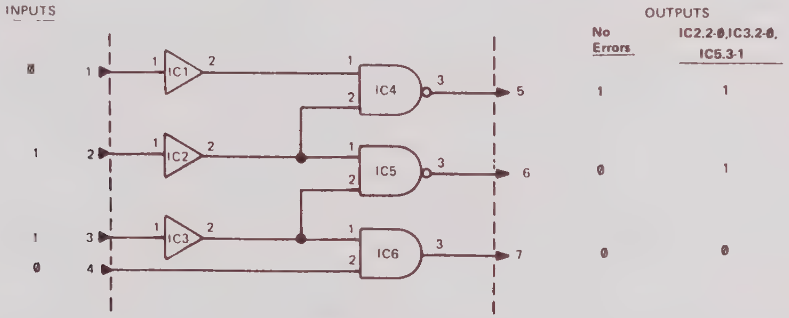


Figure 5-7. Fault Groups.

6. External Signature. A set of logical values that will be observed on circuit externals in the presence of a particular fault at the failing test step.

AFD will simulate the following faults, depending on the Fault Modeling options enabled when F is ON:

1. IC Outputs SA0 or SA1 corresponding to many output failures such as shorts to ground or V_{CC} tracks, open conditions from IC pins, or internal IC failures.

2. IC Inputs SA0 and/or SA1.

3. Shorts between adjacent IC pins caused by solder splashes or broken wire fragments.

4. User inserted failures such as a bridging failure caused by the proximity of two adjacent circuit board tracks that may have been shorted together (BLOB), or any open connections (such as faulty plated-through holes) that can occur at any location on a board (WIRE).

5. Non-functioning IC failures perhaps caused by a bad or incorrect IC, or power loss to an IC due to an open or poor connection on the board.

Faults are simulated by inserting them into the same model of the circuit that was used to determine output patterns for the good board. A fault is considered detected if for some test step the external values generated by the faulty network differ from those values generated by the good circuit.

The AFD Module, in addition to determining whether a test program can detect the failures listed, stores diagnostic files which are later used by the AFL Module in the DIAGNOSE Mode for fault analysis. Note that since the only inputs necessary to the PREPARE Mode software modules are a network description (.SR file) and a set of Input stimuli (.TP file), it is possible to generate a high quality test program for any circuit board even before the board is available.

The diagnostic (fault dictionary) files contain data that identifies the fault classes that are first detected at each test step, and the external signature of that fault class at the first failing test step. The diagnostic data is also sorted so that fault classes with identical first failing test step numbers and external signatures are grouped together in fault groups. AFD Module data outputs list the fault classes and equivalent faults modeled, and provide other fault detection data for analysis of the test program. Data outputs are described further in Para. 5.4.

5.2.6.3 Selective Fault Simulation

Selective Fault Simulation can be used to re-run selected faults after a complete fault simulation run has been completed (or at least started). This feature makes it possible to determine if a modified test program will now detect failures missed by the original test program, and it will do this in a small fraction of the time required for a complete fault simulation run. It can also be used to determine how a test program change has affected the detection of user specified detected and undetected faults associated with an IC or EXT pins.

The selective faulting feature can only be used if a normal fault simulation run (without selective faulting) has been started (or completed) previously, because the selective faulting feature requires the use of a CAPS fault dictionary file (the .FX file) that is created at the start of an AFD simulation run (F ON). If changes to the test program are also involved (and fault simulation has been run previously) then a new T and G Mode run must also be made at the same time as the selective fault run (T,G, and F ON with SF=YES). The following overall rules apply to the use of selective faulting:

1. Only the faults selected on the FAULT MODELING options page (para 5.3.5) can be selectively faulted. For example, if stuck faults at Output pins (O ON) is the only fault to be modeled during a complete fault simulation run, then only stuck faults at Output pins can be selectively faulted during the selective fault run (SF=YES).

2. If changes to the original test program (.TP file) are required, then the T, G, and F Modes must be re-run with selective faulting enabled (SF=YES) in order to determine the effect of the change on the selected faults to be modeled.

3. It is not necessary to re-run the N Mode again unless changes to the network description (.SR file) are required. If N is re-run, all CAPS/APG files will be deleted.

4. Selected faults are automatically integrated into the total fault detection data, but only for the selected faults being modeled. If the test program has been changed in the middle (as opposed to appending changes to the end of the program), it is possible that the fault detection data for other faults may be in error. Whenever possible, always append changes to the end of a test program to obtain the most accurate cumulative results.

5. If changes have been inserted into the middle of a test program, it will be necessary to re-run a complete fault simulation with selective faulting off (SF=NO) after using the selective faulting feature in order to generate an accurate data base for AFL use. If changes have been appended to the end of the program, it will not be necessary to re-run the fault simulation again.

The faults that are to be selected and run must be included in a Selective Faults file (.SF file) that can be created with EDIT. The file must be saved on the same device as the .SR file prior to starting a selective fault run, and be assigned the same name as the .SR file, but with a .SF extension. If the Selective Fault ED option is also on (ED=YES) prior to starting an AFD run , EDIT will be entered automatically prior to the run. The file can then be created and entered into core. Next, initiating an X and XMIT command from EDIT will close the .SF file, assign the correct name and extension to it, and save it on the proper device. In addition, the simulation runs selected will begin immediately without any other operator intervention.

Any number of command strings can be included in a selective faults (.SF file). The command format is listed in Table 5-4. The following command lines are typical examples of selective faults that can be run:

```
# :65,68,70;           /RUN DETECTED & UNDETECTED FAULTS 65,68,70 AGAIN
  IC:IC3;              /NEXT, RUN ALL DETECTED & UNDETECTED FAULTS
                       /ASSOCIATED WITH IC3 AGAIN
  UN EXT:30;           /NEXT, RUN UNDETECTED FAULTS ASSOCIATED WITH EXT
                       /PIN 30 AGAIN
```

Sequential selective fault classes may be specified by a dash (-) character, if desired. For example the command

```
#:65-70;
```

would allow sequential detected and undetected fault classes 65, 66, 67, 68, 69, and 70 to be run again.

When all command lines have been entered into core, the X and XMIT keys can be depressed to close the .SF file and start the run selected (if the selective fault ED=YES option has been enabled.)

If errors are detected in the .SF file as AFD is entered, execution will be suppressed, and an error message of the form:

```
ERROR AT LINE 2 OF SELFLT INPUT FILE
```

will be generated. EDIT will also be entered automatically so that the file can be corrected "on-line". Some typical .SF file error messages are listed in Table 5-5. Once the file has been corrected, and is in core, activating the

Table 5-4

SELECTED FAULT COMMAND FORMAT

Command Example	Purpose
DETECTED AND UNDETECTED FAULT SELECTION	
#: 65, 68, 70;	Run the detected or undetected faults 65, 68, and 70 again. The longer command formats NUMBER:65,68,70; OR NUMBERS:65,68,70; OR # S:65,68,70; may also be used.
IC: IC1,IC3,IC6;	Run all detected and undetected faults associated with IC1, IC3, and IC6 again. The list of ICs specified may be any length. The longer command format ICS: IC1,IC3,IC6; may also be used.
EXT: 76,25,30;	Run all detected and undetected faults associated with EXT pins 76, 25, and 30 again. The longer command format EXTS: 76,25,30; may also be used.
UNDETECTED FAULT SELECTION ONLY	
ALL UN;	Run all faults that remain undetected again. The longer command format ALL UNDETECTED; may also be specified as may any word that begins with the letters UN. Note that all command strings must terminate with a semi-colon (;).
UN IC:IC2,IC4,IC5;	Run all undetected faults associated with IC2, IC4, and IC5 again. The list of ICs specified may be any length, and the longer command format UNDETECTED ICS:IC2,IC4,IC5; may also be used.
UN EXT: 30,34,35;	Run all undetected faults associated with EXT pins 30, 34, and 35 again. The longer command format UNDETECTED EXTS: 30,34,35; may also be used.
ALL UN IC;	Run all undetected faults associated with all IC pins again. Faults concerned with EXT pins only are not included. The longer command format ALL UNDETECTED ICS; may also be specified.
ALL UN EXT;	Run all undetected faults that are associated with EXT pins again. The longer command format ALL UNDETECTED EXTS; may also be used.

Table 5-5

TYPICAL SELECTIVE FAULT ERROR MESSAGES

Message
ERROR AT LINE 7
INVALID FAULT NO.
IC NAME NOT FOUND: IC8
EXT NAME NOT FOUND: 29
NO FAULTS SELECTED FOR SIMULATION

X and XMIT keys will result in an immediate return to AFD and the run will start again at or near the point of termination. Selective fault simulation data outputs that occur as a result of a fault simulation run are described in para 5.4.

5.2.7 P Sub-Mode (Probe Files .US, .UJ, .UC, & User-Written .UD Translation).

This mode is required only if a user wishes to specify starting-point and/or jumping-point locations (including conditional-connectivity points) to the probe for use in the DIAGNOSE Mode, or if some inconsequential pulses are to be ignored by the GR pulse-catching probe. Normally, the probe will start probing at board EXT Output pins and proceed back along paths that are automatically established by the probe software as a direct result of data inputs obtained during N, T, D and G simulation runs in the PREPARE Mode. Automatic Fault Location (AFL) and the QUICK LOOKUP facility also work interactively with the probe in the DIAGNOSE Mode and will automatically furnish starting-point data to the probe that is based on an initial AFL fault evaluation or the Quick Lookup evaluation. It is also possible for a user to specify probe starting and/or jumping-point locations, if desired, or ignore inconsequential pulses by following the procedures contained in this paragraph. The P Mode is entered automatically whenever the D Mode (Figure 5-4) or Z Mode (Figure 5-13) is ON (regardless of whether P is ON or OFF). User-selected starting point, jumping points, conditional-connectivity points, and pulse-discrimination files are translated at that time, if available. The P Mode may also be entered separately by turning P ON and D and Z OFF.

User-selected starting points must be included in a source file that can be prepared with EDIT, and assigned the same name as the network description (.SR) file. The file must be located on the same device as the .SR file and have a .US extension. Starting points specify the points at which the probe can be forced to start probing under failure conditions specified by a user. They are of particular value whenever a failed board's first failing test number and/or failing output vector strongly suggest a particular failure internal to the board. Starting points may also be used whenever the UUT can be viewed as consisting of large sub-sections (or black boxes) connected by only a few common lines. The common lines may then be specified by the user as starting points for the probe. In addition, probing priorities can be established so that the order of execution of each command line in the file can be altered from the order that they were entered into the file.

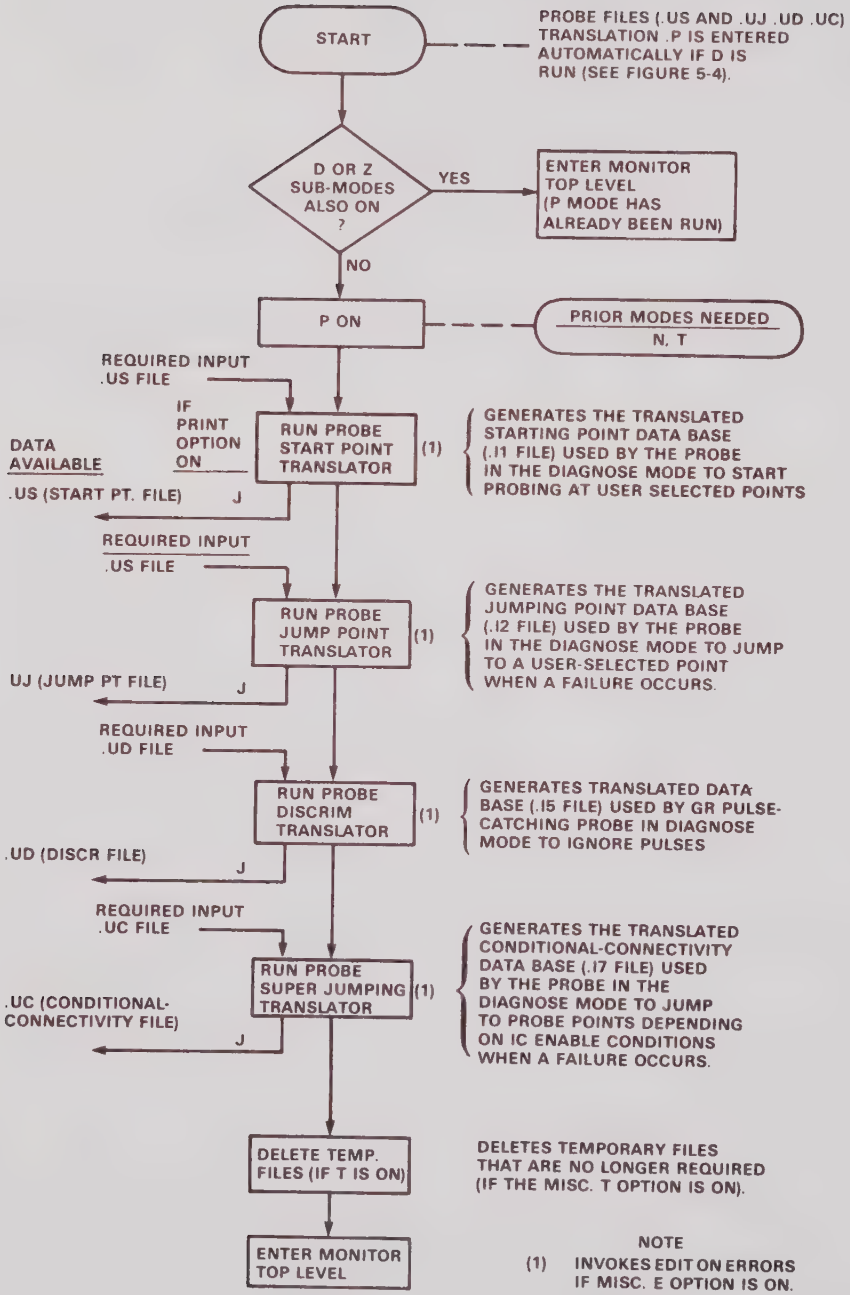

START
PROBE FILES (.US AND .UJ .UD .UC) TRANSLATION .P IS ENTERED AUTOMATICALLY IF D IS RUN (SEE FIGURE 5-4).
D OR Z SUB-MODES ALSO ON ?
YES
ENTER MONITOR TOP LEVEL (P MODE HAS ALREADY BEEN RUN)
NO
P ON
PRIOR MODES NEEDED
N, T
REQUIRED INPUT .US FILE
IF PRINT OPTION ON
DATA AVAILABLE
.US (START PT. FILE)
J
RUN PROBE START POINT TRANSLATOR
(1)
GENERATES THE TRANSLATED STARTING POINT DATA BASE (.I1 FILE) USED BY THE PROBE IN THE DIAGNOSE MODE TO START PROBING AT USER SELECTED POINTS
REQUIRED INPUT .US FILE
RUN PROBE JUMP POINT TRANSLATOR
(1)
GENERATES THE TRANSLATED JUMPING POINT DATA BASE (.I2 FILE) USED BY THE PROBE IN THE DIAGNOSE MODE TO JUMP TO A USER-SELECTED POINT WHEN A FAILURE OCCURS.
UJ (JUMP PT FILE)
J
REQUIRED INPUT .UD FILE
RUN PROBE DISCRIM TRANSLATOR
(1)
GENERATES TRANSLATED DATA BASE (.I5 FILE) USED BY GR PULSE-CATCHING PROBE IN DIAGNOSE MODE TO IGNORE PULSES
.UD (DISCR FILE)
J
REQUIRED INPUT .UC FILE
RUN PROBE SUPER JUMPING TRANSLATOR
(1)
GENERATES THE TRANSLATED CONDITIONAL-CONNECTIVITY DATA BASE (.I7 FILE) USED BY THE PROBE IN THE DIAGNOSE MODE TO JUMP TO PROBE POINTS DEPENDING ON IC ENABLE CONDITIONS WHEN A FAILURE OCCURS.
.UC (CONDITIONAL-CONNECTIVITY FILE)
J
DELETE TEMP. FILES (IF T IS ON)
DELETES TEMPORARY FILES THAT ARE NO LONGER REQUIRED (IF THE MISC. T OPTION IS ON).
ENTER MONITOR TOP LEVEL
NOTE
(1) INVOKES EDIT ON ERRORS IF MISC. E OPTION IS ON.

Figure 5-8. Major Actions Initiated when the Prepare Mode P Option is ON.

Jumping points are also included in a command file that is prepared with EDIT and assigned the same name on the same device as the network description (.SR) file, but with a .UJ extension. The jumping-point file allows a user to specify the next point to be probed if a certain point internal to a board is encountered during normal probing. The user can therefore specify the order that the probe will use as it looks at a given set of IC Input pins. A jump to a user designated point occurs only if a failure is detected (branch on errors only). Some examples of how to prepare starting-point and jumping-point files are provided in the following paragraphs, beginning with para 5.2.7.1.

A file called "conditional connectivity" can also be prepared using EDIT and assigned the same name on the same device as the network description (.SR) file, but with a .UC extension. This feature results in a reduction in the number of points which must be probed to resolve circuit faults where 2 or more IC's drive the same node. This pertains especially to devices such as bus transceivers and 3-state devices. Most 3-state and open-collector devices have chip enable or drive-enable inputs referred to as "always" inputs that can be probed first to determine if the device is driving the failing node. Only if the device is found to be driving during the failing test step will the device address, data, or other inputs be probed. This reduction in the number of points probed results in increased throughput and reduced diagnosis time. To achieve this savings in time, a user can specify a Boolean expression equating an output pin to the input drive conditions. Probe points are then indicated, based on those conditions, instead of probing all IC inputs. The .UC file is described in detail in para 5.2.7.3.

The User Pulse Discrimination (.UD) file is used only with the GR Pulse-Catching Probe. The file allows a user to resolve conflicts that can occur when a good UUT and the simulation data base mis-match because of inconsequential pulses detected by the sensitive probe. The file enables a user to specify the range of tests, modes, IC pins, etc., over which the detected pulse should be ignored.

The .UD file can be created in 2 ways; manually, or by use of the NODAL VERIFICATION (NODALV) Mode. The user can write and manually prepare .UD files by use of the EDIT mode as detailed in para 5.2.7.4. User-written Pulse Discrimination Files must be translated by running the PREPARE Mode with the option ON.

NODAL VERIFICATION automatically prepares Pulse Discrimination files based on the users network description (.SR). A "Known-Good-Board" is probed by use of the GenRad pulse-catching logic probe to establish a data base for the creation of the .UD. This data base is then automatically translated in the NODAL VERIFICATION data translator (NVTRAN) to create an IGNORE Type .UD. Other options of NODALV include the ability to clip ICs to find bad levels, and to compare .UD's to determine differences between simulation (.GV) values and the probe data base (refer to Appendix D for further details.)

After preparation, the .US, .UJ, .UC, or user-written .UD file must be translated to obtain the necessary data bases required by the probe in the DIAGNOSE Mode. Translation is accomplished by performing a PREPARE Mode run with the P sub-mode option On (see Figure 5-8) or automatically, whenever the D (or Z) sub-mode options are ON.

5.2.7.1 Preparing Starting Point (.US) Files.

Starting point files are prepared by EDIT and assigned the same name as the .SR file, but with a .US extension. Each file must be saved on the same device as the .SR file. The following command-line format may be used:

$$\underbrace{\text{START-POINT}}_{\text{IC4.5}};\underbrace{\text{PRIORITY}}_{\text{2}};\underbrace{\text{LETTER COMMAND}}_{\text{K}};\underbrace{\text{TEST NO}}_{\text{10}};\underbrace{\text{(D/S NO)}}_{\text{(8)}}\$$$

Where:

START-POINT;	Is the name (obtained from the .SR file) of the Input pin to be probed if all other conditions specified in the command string are met. If an Output pin is specified, an Input pin on the same node will be substituted, if possible. A warning message will also be generated during Probe files translation (P ON). A semi-colon (;) must terminate the argument.
PRIORITY;	Is a number from 1-7 that allows the command string to be executed in other than the order listed in the .US file. Command strings assigned the number 1 are executed first, followed by those assigned the number 2, etc. Again, the semi-colon (;) must terminate the argument.
LETTER COMMAND;	Is a letter or letters that define the conditions that must be met before the starting point specified can be used. The combinations listed in Table 5-6 are possible. Note that two letters can be combined in a single letter command as long as no space separates them. A semi-colon (;) is required to terminate the argument only if a test number or D/S specification is to follow.
TEST NO;	Defines the test number or label associated with test number commands K-P in Table 5-6. The letters D or A may be added to test numbers to specify Digital or Analog CAPS labeled tests (30:) in the .DA file. If letters are not specified, .DA file CAPS line numbers (/10/) are assumed. Test numbers do not require a terminating semi-colon (;) unless a D/S Number is to follow.
(D/S NO)	Defines the D/S name or numbers associated with D/S commands B-F in Table 5-6. Each D/S name must be enclosed in parentheses and separated by commas (2,3,4,5). A semi-colon (;) is not required.
$	Terminates the command string and must be included as the last character in each command.

Some examples of single and double-letter starting-point command lines might inlude:

Single Letter Command Line Examples

Command Line	Purpose
IC8.11;2;A$	Start probing at IC8.11 regardless of any D/S failures that occur at any test step (always start probing at IC8.11). Execution priority 2 has been assigned to this command string (command strings with priority 2 are executed ahead of those assigned priority 3, etc.) Note that D/S pin lists and test numbers are not used with A.
IC7.4;4;C;(3,4,5)$	Start probing at IC7.4 only if any combination of D/S 3,4,5 fail and no other failures occur. Execution priority 4 has been assigned to this command string.
IC8.11;3;B;(3,4,6)$	Start probing at IC8.11 only if D/S 3,4,6 all fail together and no other failures occur. Execution priority 3 has been assigned to this command string.
IC10.14;1;K;5$	Start probing at IC10.4 only if test step 5 fails. Execution priority 1 has been assigned to this command string.
IC10.11;6;L;5$	Start probing at IC10.11 only if one of test steps 1-4 fail. Execution priority 6 has been assigned to this command string.
IC7.14;7;K;5A$	Start probing at IC7.14 only if the Analog test labeled 5 fails first. Execution priority 7 has been assigned to this command string.

Table 5-6

STARTING POINT LETTER COMMANDS

Command	Function
	START REGARDLESS OF ERRORS (ALWAYS START)
A	Start probing at the point specified regardless of which error caused the board to fail (always start at the point specified).
	DRIVER/SENSOR (D/S) COMMANDS
B;(4,5,6)	The .SR file D/S numbers included in parenthesis must all fail together, but no other D/S failures can occur in order to start probing.
C;(4,5,6)	Any combination of D/S numbers listed in parenthesis can fail, but no other D/S failures can occur in order to start probing. For example, 5 can fail, or 4 and 6, or 5 and 6, etc.
D;(4,5,6)	The D/S numbers listed in parenthesis must all fail together in order to start probing. Other D/S failures may also occur.
E;(4,5,6)	Any combination of D/S numbers listed in parenthesis can fail in order to start probing. Other D/S failures may also occur.
F;(4,5,6)	Any combination of D/S numbers can fail except for the D/S numbers listed in parenthesis. The D/S numbers listed in parenthesis (4,5,6) must not fail in order to start probing.
	TEST NUMBER (LABEL) COMMANDS
K;10	Start probing if test step 10 fails.
L;10	Start probing only if the failing test step is less than test 10 (test steps 1 - 9).
M;10	Start probing only if the failing test step is less than or equal to test 10 (test steps 1 - 10).
N;10	Start probing only if the failing test step is greater than test 10 (test steps 11 - end).
O;10	Start probing only if the failing test step is equal to or greater than test 10 (test steps 10 - end).
P;10	Start probing only if the failing test step is not equal to test 10 (test steps 1 - 9 or 11 - end).

Double Letter Command Line Examples

IC7.14;5;BO;5;(7,8,9)$	Start probing at IC7.14 only if D/S 7,8,9 all fail together (but no other D/S failures occur) during one of tests 5-end. Execution priority 5 has been assigned to this command string.
IC7.14;7;FM;5;(7,8,9)$	Start probing at IC7.14 only if any set of D/S except 7,8,9 fail during one of test steps 1-5. Execution priority 7 has been assigned to this command string.

Once the .US file has been created and is in core, the EDIT X and XMIT keys can be depressed to save the file and enter the monitor top level. Next, the PREPARE Mode P option can be turned ON and a translation run started. If translation errors occur, execution will be suppressed and EDIT will again be entered (if the Miscellaneous E option is ON) so that the file can be corrected "on-line". As with the pre-processor, asterisks and up-arrows will further identify errors that occur in each command line. Once the file has been corrected and is in core, depressing X and XMIT from EDIT will automatically start the translation again. Some typical error messages that can occur are listed in Table 5-7.

Table 5-7

TYPICAL STARTING POINT AND JUMPING POINT ERROR MESSAGES

Message
D/S NOT FOUND
ILLEGAL ACTION SPECIFICATION
ILLEGAL PRIORITY
ILLEGAL SEPARATOR
ILLEGAL TEST #
IC NOT FOUND
MISSING SEPARATOR
NON-PROBEABLE POINT
PIN NOT FOUND
PREMATURE END OF COMMAND
PREMATURE END OF FILE
PLEASE REPLACE THIS PIN WITH A PROBEABLE INPUT PIN
UNKNOWN CHARACTER STRING PRESENT
WARNING: INPUT PIN SUBSTITUTED FOR GIVEN OUTPUT PIN

5.2.7.2 Preparing Jumping Point (.UJ) Files.

Jumping point files are also prepared by EDIT and assigned the same name as the .SR file, but with a .UJ extension. Each file must be saved on the same device as the .SR file. The following command-line format may be used:

```
MATCH-POINT;PRIORITY;LETTER COMMAND;JUMP TO POINT:TEST NO $

   IC8.4   ;   3   ;     K       ;     IC6.3   ;   5    $
```

Where:

MATCH-POINT;	Is the point on the board that was just probed and found to be bad. A jump to the JUMP TO POINT occurs only if the MATCH-POINT is bad. A semi-colon (;) must terminate the argument.
PRIORITY;	Is a number from 1-7 that allows the command string to be executed in other than the order listed in the .UJ file. Command strings assigned the number 1 are executed first, followed by those assigned the number 2, etc. Again, the semi-colon (;) must terminate the argument.

LETTER COMMAND;	Is a letter or letters that define the conditions that must be met before the jumping point specified can be used. The combinations listed in Table 5-8 are possible. Note that two letters can be combined in a single letter command as long as no space separates them. A semi-colon (;) must terminate the argument.
JUMP TO POINT;	This is the Input pin that is to be probed next if all conditions are met. If an Output pin is specified, an On Input pin on the same node will be substituted, if possible. A warning message will also be generated during probe files translation (P ON). The semi-colon (;) must terminate the argument.
TEST NO	Defines the test number or label associated with test number commands K-P in Table 5-8. Analog abbreviations are *not* used in jumping-point test number specifications. A semi-colon (;) is *not* used to terminate TEST NO arguments.
$	Terminates the command string and must be included as the last character in each command.

Some examples of single and double-letter jumping-point command lines might include:

Single Letter Command Line Examples

Command Line	Purpose
IC7.13;2;I;IC8.13;$	Jump to IC8.13 only if a High failure is detected at IC7.13. Execution priority 2 has been assigned to this command string.
IC7.13;4;J;IC8.13;$	Jump to IC8.13 only if a Low failure is detected at IC7.13. Execution priority 4 has been assigned to this command string.
IC7.13;3;H;IC4.3;$	Jump to IC4.3 only if an undetermined failure (neither High nor Low) is detected at IC7.13. Execution priority 3 has been assigned to this command string.
IC7.13;1;G;IC2.1;$	*Always* jump to IC2.1 whenever an error is detected at IC7.13. Execution priority 1 has been assigned to this command string.

Table 5-8

JUMPING POINT LETTER COMMANDS

Command	Function
	ALWAYS JUMP ON AN ERROR
G;	Always jump if the match point listed is bad. A semi-colon (;) must terminate the argument.
	JUMP ONLY IF ERROR PROBED IS AS SPECIFIED
H;	Jump only if the value that the probe read at the match point's failing test was undefined (could not be defined as either High or Low). A semi-colon must terminate each argument.
I;	Jump only if the value at the match point's failing test was High.
J;	Jump only if the value at the match point's failing test was Low.
	JUMP IF ERRORS ARE DETECTED AT TEST NUMBERS
K;10	Jump if test step 10 fails.
L;10	Jump only if the failing test step is less than 10 (test steps 1 - 9).
M;10	Jump only if the failing test step is less than or equal to test 10 (test steps 1 - 10).
N;10	Jump only if the failing test step is greater than test 10 (test steps 11 - end).
O;10	Jump only if the failing test step is equal to or greater than test 10 (test steps 10 - end).
P;10	Jump only if the failing test step is not equal to test 10 (test steps 1 - 9 or 11 - end).

Double-Letter Command-Line Examples

IC7.13;6;HK;IC6.2;5$	Jump to IC6.2 only if test 5 fails and the value at test 5 is undefined (neither High nor Low). Execution priority 6 has been assigned to this command string.
IC8.2;7;IL;IC4.5;3$	Jump to IC4.5 only if one of tests 1-2 fail and the value at the failing test is High. Execution priority 7 has been assigned to this command string.
IC7.13;7;JK;IC8.2;5$	Jump to IC8.2 only if test 5 fails and the value at test 5 is Low. Execution Priority 7 has been assigned to this command string.

Once the .UJ file has been created and is in core, the EDIT X and XMIT keys can be depressed to save the file and enter the top level of the Monitor. Next, the PREPARE Mode P option can be turned ON and a translation started. If translation errors occur, execution will be suppressed and EDIT will again be entered (if the Miscellaneous E option is ON) so that the file can be corrected "on-line". As with the pre-processor, asterisks and up-arrows will further identify errors that occur in each command line. Once the file has been corrected and is in core, depressing X and XMIT from EDIT will automatically start the translation again. Typical error messages that can occur are listed in Table 5-7.

5.2.7.3 Preparing Conditional-Connectivity (.UC) Files.

Conditional connectivity files are prepared by EDIT and assigned the same name as the .SR file, but with a .UC extension. Each file must be saved on the same device as the .SR file.

To take advantage of conditional-connectivity capabilities and reduce the number of probe points when 2 or more devices are driving the same node, the .UC file must provide information in 2 sections containing:

1. IC Types. A library of ICs called CCDATA.LB to which conditional connectivity can apply is supplied. The ICs in this library are listed in Table 5-9. If the IC to be included in the .UC file is in this library, no entry is necessary for it in the first section of the .UC file. For ICs that are not in the library to which conditional connectivity is applicable, an entry must be made to define each different type of IC. New elements can not be added to this library by the user.

The IC type is listed, followed by a list equating output pins to pins that are always inputs, or equating output pins to input pins qualified by a Boolean expression. An entry in this first part of the .UC file takes precedence over a CCDATA.LB library entry.

2. An entry for each output pin of each IC used on the board that is listed in the first section above. The pin entry information identifies a specific pin output and also identifies the IC type, and a list of CAPS/APG test steps during which this output pin is known to be drive-enabled.

The 2 sections of a .UC file are separated by a percent symbol (%).

First Section

If the IC type is in the library (Table 5-9), no entry for it is required in this section. ICs not included in the library must be user-defined to provide library-type information to the program. The commands for the library-type information contained in the first section have the format:

```
                       IF
IC type (output list)  or  (input list);
                       BY
```

where:

IC type is a device such as 8641, 74200, 8T26, etc, followed by either a space or a tab.

(output list) is one or more output pins, each separated by a comma or a range of pin numbers (e.g. 5-8). The list is enclosed by parentheses.

Table 5-9

CONDITIONAL CONNECTIVITY LIBRARY (CCDATA.LB)

10101
10102
10103
10104
10105
10107
10109
10110
10113
10117
10118
10131
10141
10147
10210
1802
10124
2101
2102
2901
2904
2905
2906
2907
2909
2910
2911
2915
2916
2918
3850
3870
5401
5401W
5403
5403W
5405
5405W
5406
5406W
5407
5407W
5409
5409W
5412
54125
54125W
54126
54126W
5412W
54134
54134W
54156
54156W
54159
54159W
5416
5416W
54170
54170W
54173
5417W
5422
5422W
54240
54244
54251
54365
54365W
54366
54366W
54367
54368
54368W
54374
5438
5438W
54C172
54H01
54H01W
54H05
54H05W
54H22W
54LS01
54LS01W
54LS03
54LS03W
54LS05
54LS05W
54LS09
54LS09W
54LS156W
54LS170
54LS170W
54LS173
54LS22
54LS22A
54LS241
54LS244
54LS245
54LS251
54LS251W
54LS257
54LS257W
54LS299
54LS33
54LS33W
54LS365W
54LS366
54LS366W
54LS367
54LS368
54LS368W
54LS373
54LS374
54LS38
54LS670
54S03
54S05
54S05W
54S22
54S240
54S241
54S244
54S251
54S251W
54S257
54S257W
54S299
54S373
54S374
54S734
6800
6802
6810
6820
6821
6828
6844
6845
6846
68488
6850
6852
6854
6881
7096
7097
70L96
70L97
70L98
7401
7403
7405
7406
7406W
7407
7407W
7409
7409W
7412
74125
74125W
74126
74126W
7412W
74134
74134W
74156
74156W
74159
74159W
7416
7417
74170
74173
74173
7417W
7422
74240
74241
74244
74245
74251
74257
74258
74299
7433
7433W
74363
74364
74365
74365W
74366
74368
74368W
7437
74373
74374
7437W
7438
74670
74H05
74H05W
74H22
74H22W
74L03
74LS01
74LS03
74LS03W
74LS05
74LS05W
74LS09
74LS09W
74LS15
74LS156
74LS173
74LS22
74LS22W
74LS240
74LS241
74LS245
74LS251
74LS253
74LS257
74LS257A
74LS258
74LS258A
74LS299
74LS33W
74LS363
74LS364
74LS365
74LS365W
74LS366W
74LS367
74LS368
74LS368W
74LS373
73LS374
74LS38
74LS38W
74S03
74S03W
74S134
74S173
74S189
74S22
74S22W
74S240
74S241
74S244
74S251
74S253
74S257
74S257A
74S258
74S258A
74S299
74S33
74S363
74S364
74S368W
74S373
74S374
74S38
75452
75353
75454
8035
8039
8041
8048
8049
8085
8086
8096
8097
8098
80L96
80L97
80L98
8155
8156
81LS97
81LS98
8212
8216
8226
8228
8238
8243
8251
8251A
8253
8255
8255A
8257
8259
8259A
8282
8283
8286
8287
8288
82S12
8741
8748
8755
8T26
8T96
8T97
8T98
TMS9900
7416W
74367
AM16810
CD4076
DM8599
F10131
F10231
MC10131
MC10231
MC14076
TMS9901
TMS9902
TMS9904
UHP408
UHP508
Z80 CPU
Z80 CTC
Z80 DMA
Z80 SIO
Z80 PIO

The output pin list is followed by either the keyword IF or BY and then the input-pin list. The keyword separates the output list from the input list. When the keyword BY is used the input list contains only pin numbers separated by commas. This means that the drive states of the output list are controlled BY the input-pin list. When the keyword IF is used, the input list takes the form of a Boolean expression which describes the drive states of the device in terms of pin numbers and logical operators: NOT OR NOR AND NAND XOR and ()'s. IF this expression is satisfied, the output list is considered drive-enabled. The user is then instructed to probe both the device data-input pins as well as the enabling pins. The input list is also enclosed by parentheses.

When the keyword IF is used, the operators in a Boolean expression are evaluated with respect to the following precedence:

parentheses (),

NOT

NOR and OR,

NAND and AND,

XOR,

Considerations

1. All commands are terminated by a semi-colon however, when an IC contains more than 1 group of output pins, the groups are separated by commas.

2. No commas are allowed within Boolean expressions.

Second Section

The commands of the second section in a .UC file have the format:

IC-pin-name IC-type (optional test-step list);

where:

IC-pin-name is an IC output pin (e.g. E5.6). An entry of this type is made for each output pin for which conditional connectivity is to be established. The output pin number is then followed by either a space or a tab.

IC-type is entered in the same manner as in the first section e.g. 74200. This entry is then followed by either a space or a tab.

(test-step-list) This is a list of the CAPS/APG test steps during which the output pin is drive-enabled (active). The test steps may be specified as single steps (e.g. 1, 3, 5, 17, 21), as ranges (e.g. 11-20, 35-40), or a combination thereof. The all-but (#) symbol, digital test-step labels "D" from the .DA file (e.g. 20D), and the keywords BEG (defining CAPS test step 1) and END (defining the terminating CAPS test step) can also be included (e.g. BEG-20, 35-END); however, test steps included in the list must be in ascending order.

In the second section, the optional test-step list can be included for those ICs listed in the first section with BY constructs. For those ICs that are defined with the IF construct in the first section, the test-step list is not necessary because the .UC translator evaluates the Boolean expression and then generates the test-step list automatically. If however, a test-step list is included in this case, the list takes precedence over the Boolean expression.

An Application

The Texas Instruments Inc.(1) SN74251 is an 8-to-1 Data Selector/Multiplexer with strobe-controlled 3-state outputs. Figure 5-9 is a functional logic diagram of this IC. There are 8 sources (data inputs), 3 lines for a data-select to enable one of the 8 inputs, 2 outputs (output pin 6 and its inversion - pin 5), and a strobe input (pin 7). Pin 7 (chip select) is an "always" input; the only one for this chip. The other inputs (data and select) are conditional inputs.

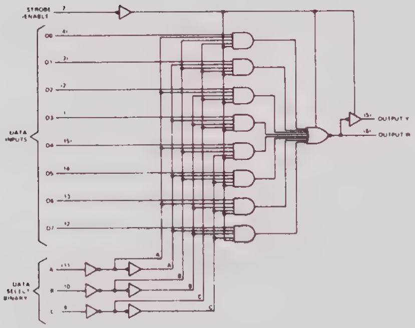
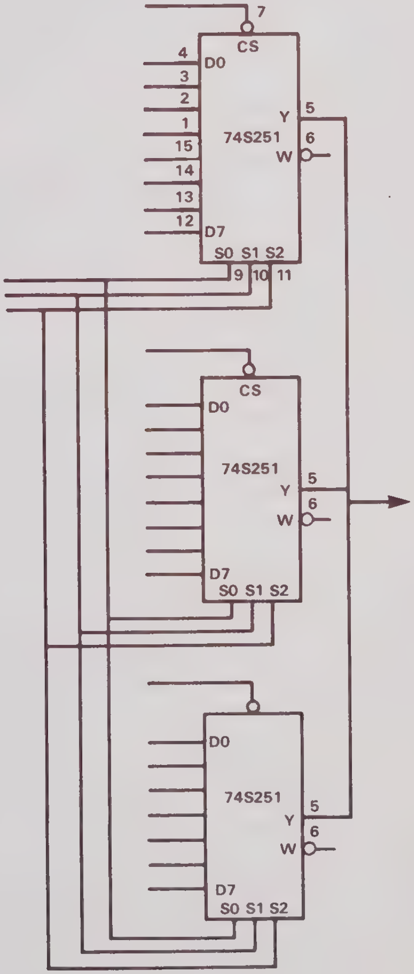

STROBE ENABLE
DATA INPUTS
D0
D1
D2
D3
D4
D5
D6
D7
OUTPUT Y
OUTPUT W
DATA SELECT BINARY
A
B
C

Figure 5-9.
SN74251 Data Selector/Multiplexers.


7
CS
4
3
2
1
15
14
13
12
D0
D7
74S251
Y
W
5
6
S0 S1 S2
9
10
11
CS
D0
74S251
5
Y
6
W
D7
S0 S1 S2
D0
74S251
5
Y
6
W
D7
S0 S1 S2

---

(1) Texas Instruments Inc. Box 5012, Dallas, Texas 75222

To enable the output (pin 6), the strobe (pin 7) must be at a low logic level. The 3-state output permits several of these ICs to be connected with the outputs connected in parallel as illustrated.

In a .UC file, a command in the first section for this IC would take the form:

74251 (5,6) IF (NOT 7); (Strobe 7 must be low to drive-enable chip).

If pin 7 is high (disabled), only pin 7 is probed. If pin 7 is low, then the normal probing acting on "stacking" all of the inputs that affect the failing output occurs. For this IC then, pins 1-4, 7, and 9-15 are probed.

Figure 5-9 illustrates 3 of these ICs with their outputs tied in parallel. Thirty pins must be probed if DRM/FINDS is not available and no conditional-connectivity capability exists. With .UC the "always" inputs are checked according to a Boolean expression and 16 pins can be probed. This eliminates unconditional probing of all of the inputs for each of the ICs.

Another Application

The DM8556 is a National Semiconductor Inc.(1) 4-bit, programmable, synchronous, binary counter with both conventional and 3-state outputs. In the high-impedance mode the 3-state outputs serve as inputs. This allows the counter to be preset to any value. Figure 5-10 illustrates this IC. The four 3-state outputs are enabled by "always" inputs 7 (logic high) and 12 (a logic low). Otherwise, the chip is in the high-impedance mode.

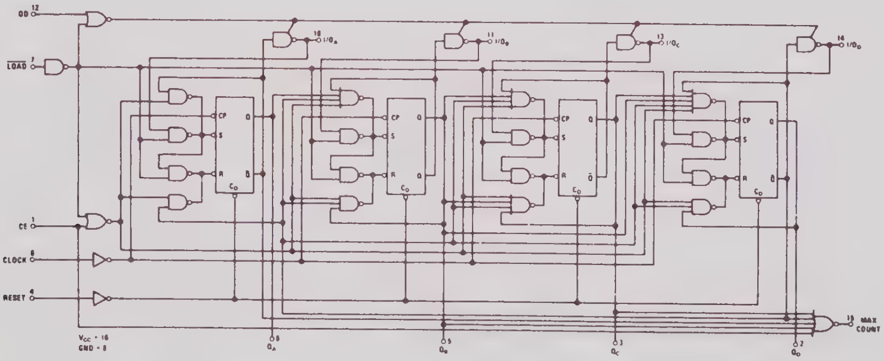


Figure 5-10. DM8556 Programmable Binary Counter.

---

(1) National Semiconductor Corp.
2900 Semiconductor Drive, Santa Clara, Calif. 95051

In a .UC file, an entry in the first section for this IC would take the form:

```
8556 (10,11,13,14) IF (7 AND (NOT 12));
```

The outputs are enabled as long as pin 7 is high (load disabled) and pin 12 (high-Z mode) is low.

If several of these chips are tied together, then a synchronous counter is formed. The maximum count output (pin 15) is tied to the chip enable input line of the next IC. The 3-state outputs are connected to a bus, and are enabled to drive the bus, one chip at a time. If a failure on a bus line is indicated, the "always" inputs 7 and 12 are checked first in an attempt to isolate the fault before the conditional inputs are checked.

A Sample .UC File

For explanation purposes, the following .UC file covers a board containing 1 or more of each of the following ICs: 8641,, 74200, 74253, 74125, 74251 and 8T26. Figure 5-11 illustrates the pin configuration for each of these ICs except for the 74251 which is illustrated in Figure 5-9.

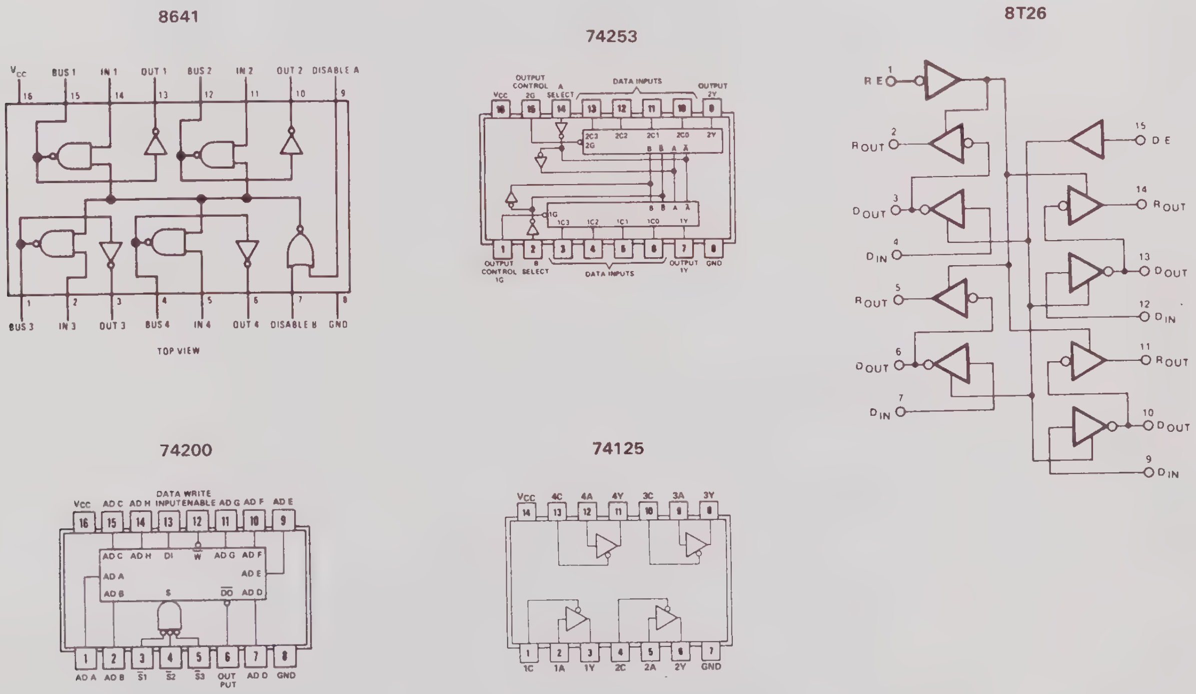

8641
V_{CC}
BUS 1
IN 1
OUT 1
BUS 2
IN 2
OUT 2
DISABLE A
BUS 3
IN 3
OUT 3
BUS 4
IN 4
OUT 4
DISABLE B
GND
TOP VIEW
74253
OUTPUT CONTROL 2G
A SELECT
DATA INPUTS
OUTPUT 2Y
OUTPUT CONTROL 1G
B SELECT
DATA INPUTS
OUTPUT 1Y
GND
8T26
RE
ROUT
DOUT
DIN
DE
74200
VCC
AD C
AD H
DATA INPUT
WRITE ENABLE
AD G
AD F
AD E
AD A
AD B
S1
S2
S3
OUT PUT
AD D
GND
74125
VCC
4C
4A
4Y
3C
3A
3Y
1C
1A
1Y
2C
2A
2Y
GND

Figure 5-11. IC Pin Configurations In SAMPLE.UC File

The first section of the SAMPLE.UC file defines the ICs shown in 5-8C. A % symbol separates the 2 sections, then the second section describes the output pins of a few of these hypothetical ICs.

```
8641     (1,4,12,15) IF (NOT(7 OR 9));

74200    (6) IF (NOT (3 OR 4 OR 5) AND 12);

74253    (7) IF (NOT(1)),
         (9) IF (NOT(15));

74251    (5,6) BY (7);

74125    (3) BY (1),
         (6) BY (4),
         (8) BY (10),
         (11) BY (13);

8T26     (3,6,10,13) IF (15),
         (2,5,11,14) IF (NOT 1);

7412     (6) BY (3,4,5),
         (8) BY (9,10,11),
         (12) BY (1,2,13);


%

E1.1     8641;
E1.4     8641;
E1.12    8641;
E1.15    8641;
E2.6     74200;
E3.7     74253;
E3.9     74253;
E4.5     74251;
E4.6     74251;
E5.3     74125    (12-17,19-END);
E5.11    74125    (8-11,18);
E7.13    8T26     (15,18,23);
E8.13    8T26     (BEG-34,74-END);
E12.12   7412     (12-17,19-100,173);
E13.6    7412     (#25-35);
E30.6    74251    (#93-END);
E31.6    74251    (15D-33D,202-END);
E32.6    74200;
E33.3    8T26;
```

NOTE

Spaces after commas are optional and can be included for readability. A tab can be substituted for any space.

Bus Contention Warning Messages

Messages similar to the following ones are warnings that may be printed on the TTY:

```
WARNING! E3.7 AND E3.9 EXHIBIT BUS CONTENTION AT TEST STEP(S): 57-59
```

This message type appears when 2 or more device outputs are enabled to drive the same node simultaneously at the test step(s) specified. Typically, this results from mis-specification of the drive-step sequence in the .TP A maximum of 6 test steps will be listed, though there may be contention in more than 6 places. The contention condition does not prevent operation. Neither the conditional-connectivity data base (.I7) file, nor the source file (.UC) is affected.

```
WARNING! A CONNECTION TO E1.1 WAS NOT FOUND IN .I7 FILE
```

This message results from the bus contention test when an IC output that is known to be on the node does not have an entry in the second section of the .UC file. Conditional connectivity cannot be effective unless all node outputs have been established. The contention condition does not prevent operation. Neither the conditional-connectivity data base (.I7) file, nor the source file (.UC) is affected.

Conditional Connectivity Translator Error Messages

Table 5-10 lists the errors that can be encountered during the translation of a .UC file.

Table 5-10

CONDITIONAL CONNECTIVITY TRANSLATOR ERROR MESSAGES

Message	Explanation
ILLEGAL SEPARATOR	
ILLEGAL CHARACTER	
MISSING SEPARATOR	
NON-EXISTENT DIGITAL TEST STEP LABEL	
ILLEGAL TEST STEP #	
ILLEGAL IC (FAKE?)	
ILLEGAL NAME	
ILLEGAL PIN #	

Table 5-10 (continued)

CONDITIONAL CONNECTIVITY TRANSLATOR ERROR MESSAGES

Message	Explanation
ILLEGAL KEYWORD	Not an IF or BY
ILLEGAL OPERATOR	NOT NOT, AND, NAND, OR, NOR, XOR
TEST STEP NUMBER DUPLICATION	
TEST STEPS OR BOOLEAN EXPRESSION NOT SPECIFIED	Test-step range takes precedence over Boolean
ILLEGAL RANGE	
PREMATURE END OF LINE	
TEST STEPS NOT IN ASCENDING ORDER	
DUPLICATE IC TYPE	
DUPLICATE PIN #	Both inputs and outputs are checked
UNMATCHED PARENTHESES	
OUTPUT PIN NOT MENTIONED IN FIRST SECTION	The IC type and pin referenced in Section 2 must be defined in Section 1 or reside in the CCDATA.LB library.
IC TYPE NOT FOUND IN .UC OR LIBRARY	

5.2.7.4 Manually Preparing a .UD File.

Pulse Discrimination Files are automatically created by use of the Nodal Verification Mode. If, however, a user wants to manually prepare a .UD, follow the instructions as outlined in the following paragraphs. For further information on Nodal Verification refer to Appendix D.

The user written pulse discrimination file is prepared in EDIT. The file is assigned the same name as the .SR but with a .UD extension. The file must be saved on the same surface as the .SR file. The following command line format may be used.

$$\overbrace{\text{ENABLE;}}^{\text{FILE TYPE}}$$

$$\overbrace{\text{NODE IC2.1}}^{\text{NODAL DATA}}\ \overbrace{\text{(20-40,60A,70D)}}^{\text{TEST STEP/RANGE}}\ \overbrace{\text{ILLEGAL;}}^{\text{ERROR TYPE}}$$

Where:

FILE TYPE	Is the type of .UD file to be used. The following file types may be specifid as the first file entry:

ENABLE; Use mis-match errors for the nodal data and test step/ranges included in the file only. Ignore all other discrepancies.

IGNORE; Ignore mis-match errors for the nodal data and test step/ranges included in the file only. Use all other discrepancies.

If a file type (ENABLE or IGNORE) is not listed as the first file entry, IGNORE is assumed. The semi-colon (;) terminates the command string.

NOTE

NODAL VERIFICATION cannot make or add to an ENABLE type .UD. Use of (user written) ENABLE-files in NODAL VERIFICATION applications will be limited (refer to Appendix D).

NODAL DATA — Is the node, IC, IC pin, Driver/Sensor, or Allnodes over which the ENABLE or IGNORE errors command is to apply. The following format may be used:

IC8	IC8 only.
IC4.5	IC4 pin 5 only.
NODE IC2.1	The node that includes IC2 pin 1. At least one space must be inserted between the NODE command and the IC or Driver/Sensor designation that follows.
NODE 8	The node that includes Driver/Sensor 8.
NODE AB 5	The node that includes the Driver/Sensor named AB5. Note that Driver/Sensor names which are numbers will be interpreted as numbers. For example, if the .SR file lists the name as Driver/Sensor 3, then the number 3 will be used even though the coding might be: 3 EXT 4;
8	Driver/Sensor 8 only.
ALLNODES	All nodes.

TEST STEP/RANGE — Is the test step(s) and/or range of test steps over which the ENABLE or IGNORE errors command is to apply. The letters D or A may be added to specify Analog or Digital test step labels (30:) in the .DA file. If letters are not specified, .DA file CAPS line numbers (/10/) are *assumed*. The following format may be used.

(10)	Line number 10 in the .DA file only.
(10,35,48)	Line numbers 10, 35, and 48 only.
(#20,40)	All but line numbers 20 and 40.
(20A,40A,50D,70D,80)	Analog test steps 20 and 40. Digital test steps 50 and 70, and line number 80 in the .DA file.
(#20,30D-60D,70)	All but Analog Test step 20. Digital test steps 30 *through* 60, and line number 70 in the .DA file.

Test step(s) and/or test step ranges must be included in parentheses, with each test step or test step range separated by a comma. The dash (-) indicates *through*. It is also possible to specify *all* test steps or *no* test steps as the following examples indicate.

(ALL)	*ALL* CAPS test steps.
(NONE)	*No* CAPS test steps.

A range of test steps that includes every step from the test step specified through to the *end* of the test program may also be listed as the *following* example indicates:

(40D-END) Digital test step 40 through to the *end* of the CAPS section.

ERROR TYPE

One of the following error types *must* be selected:

MISSING;	A missing pulse (the CAPS simulator has detected a pulse that the probed did *not* detect).
ILLEGAL;	An illegal pulse (the probe has detected a pulse that the CAPS simulator did not detect).
HALFP;	A mid-range half-pulse has been detected by the probe. Since the CAPS simulator does not simulate half-pulses, an error will occur. Predicted errors of this type can be ignored through use of the .UD file.

The error type is the last entry in the command string. The semi-colon (;) terminates the string.

The following general rules should be followed when preparing any .UD file with EDIT:

1. Always list the FILE TYPE first (ENABLE or IGNORE). If a FILE TYPE is not listed, IGNORE is assumed.

2. When listing NODAL DATA, an IC name that does not include a pin number (IC20) cannot be specified as part of a NODE. IC20 means IC20 only. If an IC pin number is included (IC20.8) then the IC can be listed by itself or as part of a node as the following 2 examples indicate:

NODE IC20.8	The node that includes IC20 pin 8.
IC20.8	IC20 pin 8 only.

3. If a pin has been specified in a command string for one error type (MISSING, ILLEGAL, or HALFP), it cannot be specified again for the same error type. For example, the command strings

```
IC10.2(20D,40D)MISSING;
IC10.2(30A,80A)MISSING;
```

are illegal as the same IC pin number has been specified twice for MISSING, and the two command strings could have been combined into the single command string:

```
IC10.2(20D,30A,40D,80A)MISSING;
```

4. The ALLNODES command implies all nodes. The command, will, however, be overridden by any other command strings in the file that apply to the same error type (MISSING, ILLEGAL, or HALFP) as ALLNODES. For example, the command strings

```
IGNORE;
ALLNODES(ALL)ILLEGAL;
ALLNODES(ALL)HALFP;
IC4.5(#20D)HALFP;
```

specify that all illegal pulses and all half-pulses are to be ignored (turned OFF) with the exception of half-pulses that occur at IC4.5 in test step 20D. The command strings also imply that all missing pulses are to be used (turned ON) as are half-pulses if they occur at test step 20D when probing IC4.5.

5. The NODE command applies to any wired-OR connections that are attached to the node specified, however, the GR Pulse-catching probe cannot be used to cross over any simulated FAKE elements on the node.

6. Test steps and ranges of tests must always start with the lowest test number in the list. The remaining numbers can be in any order as the following examples indicate.

```
(10,40,20,80,30)
(10D-40D,50D,80A,60D)
```

The following examples of valid .UD file command strings illustrate the format that may be used:

```
ENABLE;                                IGNORE;
NODE IC24.1(20D,40D)MISSING;           ALLNODES(ALL)MISSING;
IC10.2(10-30,40A)ILLEGAL;              ALLNODES(ALL)ILLEGAL;
IC3.4(#28D-46D)HALFP;                  ALLNODES(ALL)HALFP;
NODE IC2.8(ALL)HALFP;                  NODE IC24.1(#20D,40D)MISSING;
                                       IC10.2(#10-30,40A)ILLEGAL;
                                       IC3.4(28D-46D)HALFP;
                                       NODE IC2.8(NONE)HALFP;
```

The command strings listed will enable (turn ON) and ignore (turn OFF) the following pulses:

Pulse	ON	OFF	Probed Point
MISSING	20D,40D	All Else	NODE IC24.1
ILLEGAL	10-30,40A	All Else	IC10.2
HALFP	All Else	28D-46D	IC3.4
HALFP	All	None	NODE IC2.8

Note that the function of each command string in either file is identical in the examples shown. Either file (ENABLE or IGNORE) may be prepared depending on whether the greater number of pulse errors are to be enabled or ignored. For example, if half-pulses are to ignored on 6 different nodes (out of a possible total of 8 nodes), it might be easier to prepare an enable file as the following examples illustrate:

```
IGNORE;                       ENABLE;
NODE IC2.1(ALL)HALFP;         ALLNODES(ALL)MISSING;
NODE IC3.2(ALL)HALFP;         ALLNODES(ALL)ILLEGAL;
NODE IC4.5(ALL)HALFP;         NODE IC8.6(NONE)HALFP;
NODE IC5.3(ALL)HALFP;         NODE IC9.3(NONE)HALFP;
NODE IC6.4(ALL)HALFP;
NODE IC7.2(ALL)HALFP;
```

The IGNORE file required 6 command strings in order to ignore 6 of the possible 8 nodes, while the ENABLE file accomplished the same function with 4 command strings.

Any number of command strings can be included in the .UD file. Once the file has been created and is in core, type X and XMIT to save the file and enter the Monitor top level. Next, the PREPARE Mode P option can be turned ON and a translation run started. If translation errors occur, execution will be suppressed and EDIT will again be entered (if the MISC E option is ON) so that the file can be corrected "on-line". Asterisks and up-arrows will further identify errors that occur in each command line. Once the file has been corrected and is in core, depressing X and XMIT from EDIT will automatically start the translation again. Some typical error messages that can occur include:

Error	Description
1	ILLEGAL SEPARATOR
2	ILLEGAL CHARACTER
3	MISSING SEPARATOR
4	LOGICAL END OF COMMAND EXCEEDED
5	ILLEGAL D/S NUMBER
6	ILLEGAL IC (FAKE?)
7	ILLEGAL NAME
8	ILLEGAL PIN NUMBER
9	ILLEGAL LABEL
10	TEST STEP NUMBER DUPLICATION
11	DUPLICATE MISSING, ILLEGAL, HALFP SPECIFICATION
12	ILLEGAL RANGE
13	PREMATURE END OF COMMAND
14	PREMATURE END OF FILE
15	LOWEST TEST STEP IS NOT FIRST TEST STEP

5.2.8 Q Sub-Mode (Quick Lookup Preparation).

This sub-mode is required only if it is desired to use the Quick Dictionary Lookup Option in the DIAGNOSE Mode to pass probe starting points, resulting from a Quick Fault Dictionary Lookup, to the probe. This method may prove significantly faster then diagnosing faults with AFL enabled if a user does not intend to perform a complete AFL run; but instead, wishes to pass starting points to the probe from an initial AFL Lookup only. The Quick distionary lookup cannot be used if the DIAGNOSE Mode AFL Option is ON.

The Q sub-mode of PREPARE (Figure 5-12) can only be run after completing an N, T, G, and F run in order to generate the needed data base files. When Q is run, a QL data base file is created. The file is needed in order to run the DIAGNOSE Mode with the Quick Dictionary Lookup enabled and Quickly pass starting points to the probe.

5.2.9 Z Sub-Mode (Percent Processing Update).

This sub-mode is used to update percent processing commands withough rerunning the entire PREPARE Mode again. It is also used to complete the conversion of CAPS VII data base files to CAPS IX once the CONVERT Mode has been run. Details are provided in the following paragraphs.

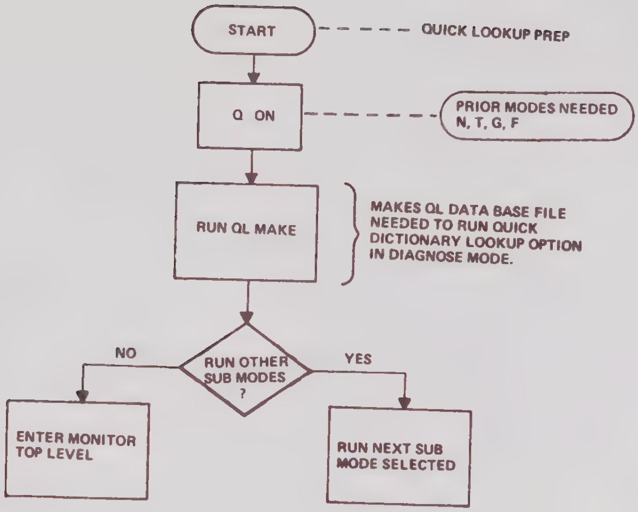

START
QUICK LOOKUP PREP
Q ON
PRIOR MODES NEEDED N, T, G, F
RUN QL MAKE
MAKES QL DATA BASE FILE NEEDED TO RUN QUICK DICTIONARY LOOKUP OPTION IN DIAGNOSE MODE.
RUN OTHER SUB MODES ?
NO
YES
ENTER MONITOR TOP LEVEL
RUN NEXT SUB MODE SELECTED

Figure 5-12. Major Actions Initiated When The Prepared Mode Q Option is ON.

5.2.9.1 Percent Processing Updates

It may sometimes be desirable to update (change) selected percent processing commands in the .SR file or include a percent processing command that was not previously there. For example, during board testing in the DIAGNOSE Mode, the GUIDED PROBE N option can be turned ON to use the BED-OF-NAILS feature so that all EXTernal pins that are connected to test system sensor pins (except for those pins specified in the .SR file as NONAIL pins) will not have to be probed. As the test sequence is run, it may be desirable to include an additional pin as a NONAIL pin or remove the NONAIL command from a pin. The Z sub-mode of PREPARE can be used for this purpose without having to re-run the entire PREPARE Mode again.

Changes to the percent processing commands can be made rapidly by re-entering the monitor top level, editing the changes into the .SR file, and then running the changed .SR file through the Z sub-mode of PREPARE. The following Percent Processing Commands can be changed in this manner:

PROBE	(Removes NOPROBE Command)
PRINT	(Removes NOPRINT Command)
CLIP	(Removes NOCLIP Command)
NONAIL	(Insert NONAIL Command)
THRESH	(Set THRESHold Levels Command)
POWER	(Define POWER pin connections for FINDS)
MESSAGE=	(Displays PROBE Message)
LOC=	(Prints LOCATION Message)
UV	(Removes NOUV Command)
CLIPD	(Allows use of the DRM for clipping)
PROBED	(Allows use of the DRM for probing)
NOOHM	(Ignore ohmic measurements)

The purpose of each Percent Processing Command together with methods of inserting it into the .SR File are described further in Para. 2.3.5. Once the .SR File has been corrected, the PREPARE Mode can be entered, and the Z sub-mode turned ON. When the Z sub-mode is run, the following major actions take place (see Figure 5-13):

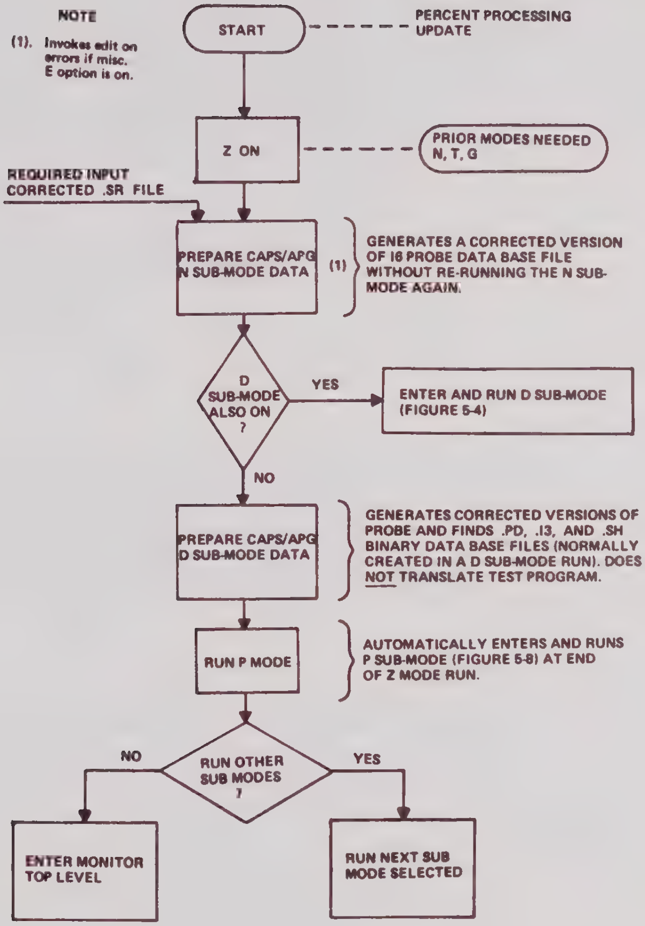

NOTE
(1). Invokes edit on errors if misc. E option is on.
START
PERCENT PROCESSING UPDATE
Z ON
PRIOR MODES NEEDED N, T, G
REQUIRED INPUT CORRECTED .SR FILE
PREPARE CAPS/APG N SUB-MODE DATA
(1)
GENERATES A CORRECTED VERSION OF I6 PROBE DATA BASE FILE WITHOUT RE-RUNNING THE N SUB-MODE AGAIN.
D SUB-MODE ALSO ON ?
YES
ENTER AND RUN D SUB-MODE (FIGURE 5-4)
NO
PREPARE CAPS/APG D SUB-MODE DATA
GENERATES CORRECTED VERSIONS OF PROBE AND FINDS .PD, .I3, AND .SH BINARY DATA BASE FILES (NORMALLY CREATED IN A D SUB-MODE RUN). DOES NOT TRANSLATE TEST PROGRAM.
RUN P MODE
AUTOMATICALLY ENTERS AND RUNS P SUB-MODE (FIGURE 5-8) AT END OF Z MODE RUN.
RUN OTHER SUB MODES ?
NO
YES
ENTER MONITOR TOP LEVEL
RUN NEXT SUB MODE SELECTED

Figure 5-13. Major Actions Initiated When The Prepare Mode Z Option is ON For Percent Updates.

5.2.9 I Sub-Mode (Incremental TP Prep/Simulation).

When the I sub-mode of PREPARE is turned ON, all other sub-modes (N,T,G,D, F,Q,P,Z) are automatically turned OFF. This sub-mode allows test programs to be prepared and checked against the simulated model in increments. Each increment can then be accepted or rejected depending upon the outcome of the incremental simulation. The N sub-mode of PREPARE must have been run prior to using the I sub-mode.

The I Sub-Mode of PREPARE is available with APG equipped test systems only. Methods of using the I Sub-Mode are described in Appendix B.

5.3 PREPARE MODE OPTIONS

5.3.1 General

All of the optional commands necessary to allow efficient operation within the PREPARE Mode are contained on six separate pages if the CAPS Monitor is in use, and seven pages if the APG Monitor is being used. Methods of changing page options, advancing pages, etc., have been described in Section 3 (Operation). The following paragraphs define the PREPARE Mode optional commands available and briefly define their use.

5.3.2 File Usage Options

The FILE USAGE Page is the first page displayed when the PREPARE Mode is entered from the CAPS or APG Monitors. The page provides a means of listing all of the file name and device data required to perform a PREPARE Mode run (see Figure 5-14). The commands described in the following paragraphs may be issued from this page.

5.3.2.1 The Network File Name and Device (N=).

This is the name assigned to the network description (.SR) file that is to be run in the PREPARE Mode and the device on which the file is located. Files may be located on any file structured disk based device. If a device is not specified, the default device (DSK:) is assumed. If an extension is not specified, the .SR extension is assumed. A file labeled NAMEA.SR and located on DSK: could therefore be specified by the simple command:

```
N=NAMEA
```

If the file had been located instead on DSK2:, the following command would have been necessary:

```
N=DSK2:NAMEA
```

5.3.2.2 The Test Program File Name and Device (T=).

This is the name assigned to the basic test program (.TP) file that is to be run in the PREPARE Mode, and the device on which the file is located. Files may be located on any file structured disk based device. If a device is not specified, DSK: is assumed. If an extension is not specified, the .TP extension is assumed. A file labeled NAMEB.TP and located on DSK: could therefore be specified as:

```
T=NAMEB
```

If the file had been located on DSK2: the following command line would have been necessary:

```
T=DSK2:NAMEB
```

5.3.2.3 User IC Library File (U=).

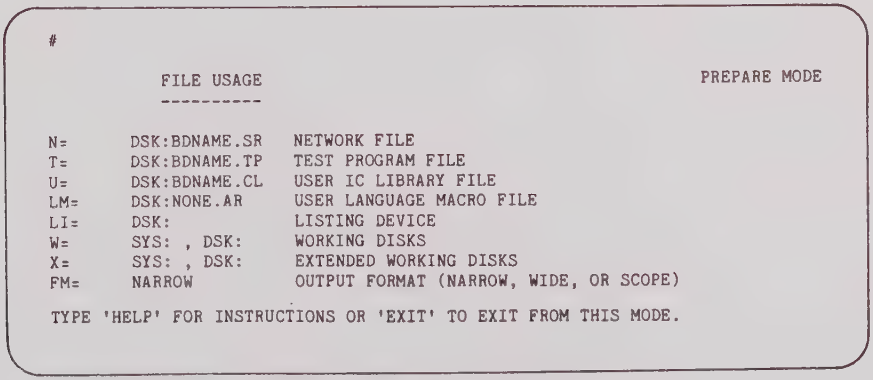
This is the name assigned to the CAPS User Library (the CAPS Library that contains user inserted devices). During library generation in the GENERATE Mode, user selected devices may be entered into a file on the DSK: device identified as DSK:USERLB. During a PREPARE Mode run, user devices in that file will be simulated if coded into the network description .SR file in the normal manner, and listed under U= on the FILE USAGE page as in the following example:

```
U=DSK:USERLB.CL
```

An extension assignment (.CL) is not mandatory, however, use of the .CL extension is recommended for proper file management. The use of the device name (DSK:) also is not necessary, as DSK: is assumed.

If another name or device is to be assigned to DSK:USERLB.CL, then the CCL .COPY command can be used from the OS/8 keyboard monitor to change the name and/or device location as in the following example:

```
.COP DSK2:SPECLB.CL<DSK:USERLB.CL
```



```
#

          FILE USAGE                                       PREPARE MODE
          ----------

N=      DSK:BDNAME.SR    NETWORK FILE
T=      DSK:BDNAME.TP    TEST PROGRAM FILE
U=      DSK:BDNAME.CL    USER IC LIBRARY FILE
LM=     DSK:NONE.AR      USER LANGUAGE MACRO FILE
LI=     DSK:             LISTING DEVICE
W=      SYS:  , DSK:     WORKING DISKS
X=      SYS:  , DSK:     EXTENDED WORKING DISKS
FM=     NARROW           OUTPUT FORMAT (NARROW, WIDE, OR SCOPE)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-14. The Prepare Mode FILE USAGE Page.

This example creates the file SPECLB.CL on DSK2:. Next, the PREPARE Mode can be entered and a new device name specified as follows:

```
U=DSK2:SPECLB.CL
```

5.3.2.4 User Language Macro File (LM=).

This option specifies the Archive file name, extension, and device location. The Archive file may contain user-generated Macros that can be automatically inserted into the test program and expanded by the Macro Processor as the program is translated in the T sub-mode of PREPARE (refer to Para. 5.2.3). If a device is not specified for the Archive file, DSK: is assumed. If an extension is not specified, the .AR extension is assumed. A file labeled NAMEA.AR and located on DSK: could therefore be specified by the command:

```
LM=NAMEA
```

If the Macro cannot be located in the user-generated Archive File specified by the LM= Option, an automatic search of any GR supplied system macro Archive File (DSK:MACLIB.AR) will occur next. If the Macro cannot be located in either Archive File, an UNDEFINED MACRO Message will be output.

5.3.2.5 Listing Device (LI=).

Output all CAPS PREPARE Mode listings to the device specified as in the following examples:

LI=LPT:	Output listings to the line printer.
LI=TEC:	Output listings to the TEC data terminal display.
LI=TTY:	Output listings to the Teletype.

It is also possible to suppress all PREPARE Mode data outputs by listing DSK: as the Output device. For example,

```
LI=DSK:
```

will output nothing (suppress data outputs). All CAPS output files are always saved on disk regardless of the listing device specified.

5.3.2.6 Working (W=) And Extended Working (X=) Disks.

During a PREPARE Mode run, both CAPS and APG create a series of special-purpose test (binary), listing (ASCII), and intermediate files. The testing files are directed to the W= disk specification. The listing and intermediate files are directed to the X= specification. On dual-drive disk systems the W= and X= specifications are typically the same; on systems with more than two drives, the W= and X= specifications can be, and usually are different. Table 5-11 lists the CAPS/APG files created during a PREPARE Mode run, briefly defines their purpose, and lists normal file locations (for X= and W= DSKm:, DSKn:), where m and n are single digits from 0-7 depending on the disk drive configuration. Temporary files listed in the table are CAPS/APG files that may be deleted at the end of a run, or are deleted automatically by CAPS/APG during the run.

The following are common disk configurations:

Table 5-11

CAPS AND APG PREPARE MODE FILES

Ext.	Name	Sub-Mode Made by	Sub-Mode Used by	Type File	Location(1) DKSm:	Location(1) DSKn:
.BE	CAPS Trans Errors	T	T	Listing	X	
.BR	Bad					
.CH	Fault Dictionary File	F	F,Q	Testing	W	
.CM	AFD/AFL Compare File	G	F,Q	Testing	W	
.CT	D/S to CAPS Ext Map	N	Q	Testing	W	
.DA	CAPS Auto-Prog TP	G	Dd	Listing		W
.DF	Difference Report	G	G	Listing	X	
.ET	Expanded Macro TP(3)	T	T,G	Listing		X
.EW	Pre-Proc Errors	N	N	Listing	X	
.FC	Fault Dictionary File	F	F,Q	Testing		W
.FG	Fault Gp & % Det.	F	F	Listing		W
.FO	Fault Dictionary File	F	F,Q	Testing		W
.FP	Detected Faults List	F	F	Listing	W	
.FQ	Sel Fault Temp. .FX	F	F	Temporary	W	
.FS	Sel Fault Temp. .FO	F	F	Temporary	W	
.FT	Master Fault Table	F	F	Testing		W
.FU	AFD Fault Table	F	F,Q	Testing	W	
.FX	Fault Dictionary File	F	F	Temporary	W	
.GO	Good Sim Outputs	G	G	Temporary		W
.GR	Good Results	G	G	Temporary		X
.GV	Probe Good Values	G	G,D,F,Z	Testing	W	
.HB	2225 High-Speed Binary		*	Testing		
.HD	CAPS Header	N,T	N,T	Listing	X	
.IB	2225 Image Binary		*	Testing		
.IC	2225 Image Neglect	2225		Temporary	W	
.IM	2225 Image File	2225		Testing/Listing		W
.IN	CAPS Trans Binary	T	T,G,F	Temporary		W
.IS	2225 Image Signature	2225		Temporary		W
.IV	2225 Image Neglect	2225		Temporary	W	
.IW	2225 Image Warnings	2225		Listing	W	
.I1	Start Pt Data Base	P,D,Z	*(2)	Testing		W
.I2	Jump Pt Data Base	P,D,Z	*	Testing		W
.I3	IC Name Data Base	D,Z	D,Z	Testing		W
.I5	User Discr. Data Base	P,D,Z	*	Testing	W	
.I6	% Proc. & Msg. Data Base	N,Z	N,F	Testing		W
.I7	Conditional Connectivity Data	P,D,Z	*	Testing		W
.LT	Label Table	T	T,D,P,Z	Testing		W
.ME	Macro Errors	T	T	Listing	X	
.NM	IC & EXT Names	N	G,D,F,Z,P	Testing		W
.PD	Probe Data File	D,Z	*	Testing		W
.PN	Pin Name File	N	T	Temporary	W	
.PX	Temp .US,.UJ,.UC,.UD with Errs.	D,P,Z	D,P,Z	Temporary		W
.QL	QL Fault Dict. File	Q	*	Testing	W	
.SH	DRM Sim. Data Base	D,Z	*	Testing	W	
.SO	SF Dictionary File	F	F	Temporary		W
.SP	SF Det. Faults .SU to .FT Map	F	F	Listing	W	
.SS	SS Dictionary File	F	F	Temporary	W	
.ST	Sim Data Base	N	N,T,G,F, P,D,Q,Z	Testing		W
.SU	AFD Sel Faults Table	F	F	Temporary	W	
.SX	Temp. .SR With Errors	N,Z	N,Z	Temporary	W	
.TA	Test Binary (1792A/B)	D	*	Testing		W
.TD	Test Binary (1792C/D)	D	*	Testing		W
.TL	Test Binary (1795/1799)	D	*	Testing		W
.TV	Test Binary (1796)	D	*	Testing		W
.TW	2225 Test Program Warnings	2225		Listing	W	
.TZ	2225 Test Program	2225	*	Testing/Listing		W
.WN	Warnings & Statistics	N	N	Listing		X
.XR	Cross-Ref. Data Base	N	N	Listing		X
.XX	Temporary	N	N*	Temporary	W,X	W,X

FILE TYPES

Listing files are ASCII source listing files.
Testing files are binary testing files.
Temporary files are files that are normally deleted by the system software.

NOTE

The following file extensions are reserved for CAPS/APG users:

All extensions beginning with a number;
All 2 letter extensions of the form:
.AA, .BB, .CC, etc.
with the following exceptions:
Screen.AA
*.SS
*.XX

---

(1) W = Working Disks, X = Extended Working Disks.
(2) * = Binary testing files used in DIAGNOSE Mode also.
(3) The .ET has been moved to the extended working disks if they are different from the working disks. This allows longer .ET and .DA files.

System	Configuration	Contents
1-DRIVE	W=DSK0:,DSK1: X=DSK0:,DSK1:	testing files and listing files on the same disk.
2-DRIVE	W=DSK2:,DSK3: X=DSK2:,DSK3:	testing files listing and intermediate files
2-DRIVE	W=DSK2:,DSK3: X=DSK0:,DSK1:	testing files listing and intermediate files DSK1: provides additional space
3-DRIVE	W=DSK2:,DSK3: X=DSK4:,DSK5:	testing files listing and intermediate files

NOTE

When a W= or X= specification is made for any mode, the change is automatically made, and applies to all applicable modes.

5.3.2.7 Output Format (FM=).

Selects the proper output format for listings (NARROW, WIDE, or SCOPE). Restricts the total character width and length that can be output as follows:

Command	Maximum Width	Maximum Length
SCOPE	72 characters	24 lines
NARROW	72 characters	60 lines
WIDE	128 characters	60 lines

If TEC: data terminal display outputs are desired, always specify SCOPE.

```
FM=SCOPE
```

If Teletype data outputs are desired (or line printer with 8½x11" paper), use NARROW.

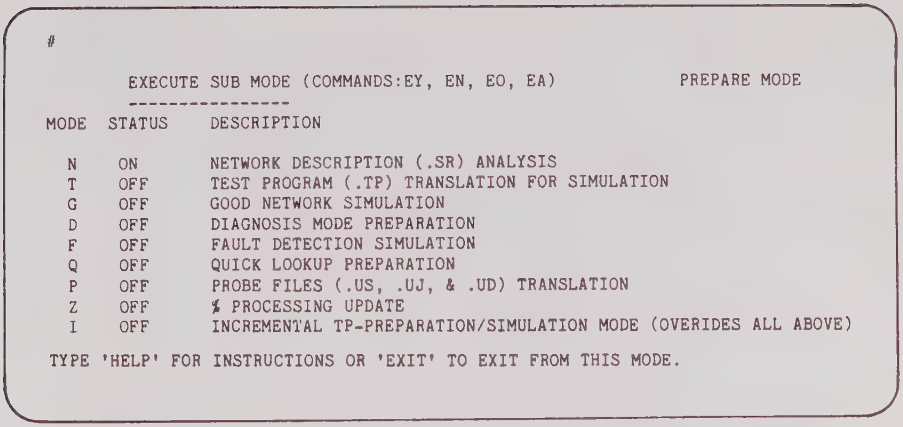
```
FM=NARROW
```

If wide paper line printer data outputs are desired, either NARROW or WIDE will work, however, WIDE is recommended for the most efficient utilization of printer paper.

```
FM=WIDE
```

5.3.3 Execute Sub Mode Options

The EXECUTE SUB MODE page is the second page that can be displayed. The page contains a series of sub mode letter switches (N,T,G,D,F,Q,P,Z,I) that turn individual sections of the CAPS or APG software ON or OFF to accomplish specific tasks during a PREPARE Mode run (see Figure 5-15). The tasks performed by each sub-mode switch are described in Para. 5.2.1 through 5.2.10. Briefly, the activation of each switch has the following effect:



```
#

       EXECUTE SUB MODE (COMMANDS:EY, EN, EO, EA)           PREPARE MODE
       ----------------
MODE  STATUS    DESCRIPTION

  N    ON       NETWORK DESCRIPTION (.SR) ANALYSIS
  T    OFF      TEST PROGRAM (.TP) TRANSLATION FOR SIMULATION
  G    OFF      GOOD NETWORK SIMULATION
  D    OFF      DIAGNOSIS MODE PREPARATION
  F    OFF      FAULT DETECTION SIMULATION
  Q    OFF      QUICK LOOKUP PREPARATION
  P    OFF      PROBE FILES (.US, .UJ, & .UD) TRANSLATION
  Z    OFF      % PROCESSING UPDATE
  I    OFF      INCREMENTAL TP-PREPARATION/SIMULATION MODE (OVERIDES ALL ABOVE)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

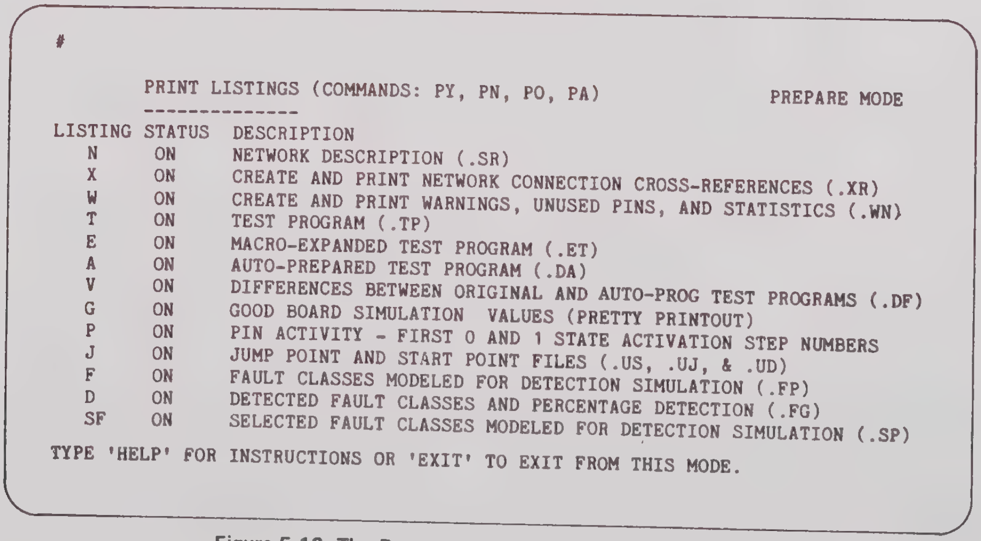
Figure 5-15. The Prepare Mode EXECUTE Page.

<u>Mode Switch ON</u>	<u>Function</u>
N	For network description (.SR) analysis. Checks the basic network description (.SR file) for syntax, semantic, and other errors. Also generates cross-reference data, warnings, and statistics (refer to Para. 5.2.2). Convert alias IC names.
T	For test program (.TP) translation for <u>simulation.</u> Expands test program macro commands (if any), checks the test program for translator defined errors, and generates a CAPS/APG test program data base needed by other major modes during a PREPARE Mode run (refer to Para. 5.2.3).
G	For good network simulation. Simulates a good UUT. Adds outputs to the test program and compares them against expected outputs. Creates a digitally auto-programmed test program and generates both good board simulation results and pin activation data for test program analysis (refer to Para. 5.2.4).
D	For Diagnosis Mode preparation. Creates binary data bases used by both the probe and FINDS, and a test program binary file used during UUT testing in the DIAGNOSE Mode. Also causes the P Mode to be run automatically (refer to Para. 5.2.5).
F	For Fault Detection Simulation. Identifies and collapses detectable failures into fault classes, simulates faulty networks, and compares test program outputs against fault simulated outputs (refer to Para. 5.2.6).
Q	For Quick Dictionary Lookup Preparation. Prepares the .QL Binary Data Base File required for use by the Quick Dictionary Lookup option in the DIAGNOSE Mode. The Quick Dictionary Lookup passes probe starting points, obtained from a Quick Fault Dictionary Lookup, to the probe (refer to Para. 5.2.8).

P	For Probe files (.US, .UJ, .UC, & .UD) Translation only. Generates translated probe starting point and jumping point data bases, a User Pulse Discrimination data base and a User Conditional-Connectivity data base. This mode is not required for .UD files created by NODAL VERIFICATION (refer to Appendix D), because these files are automatically translated in NODAL VERIFICATION. This mode is required only if a user wishes to specify probe starting and/or jumping points or pulses that are to be ignored by the GenRad pulse-catching probe. It is not required for normal probe operation (refer to para 5.2.7).
Z	For Percent Processing Updates. Also updates CAPS VII data base files to CAPS IX. Updates selected percent processing commands (such as NONAIL, CLIP, THRESH) in the Network Description (.SR File) without having to re-run the entire PREPARE Mode again (refer to Para. 5.2.9).
I	For Incremental Test Program Preparation and Simulation. Allows test programs to be prepared and checked against the simulated model in increments which can then be accepted or rejected (refer to Appendix B. When this sub-mode is ON, all other sub-modes (N,T,G,D,F,Q,P,Z) are automatically turned OFF. The I sub-mode is available only with APG.

5.3.4 Print Listings Options

The PRINT LISTINGS page is the third page that can be displayed. The page contains a series of letter switches to allow user selection of PREPARE Mode listings on an "as needed" basis (see Figure 5-16). Each listing is described in Para. 5.4. The following listings may be obtained as each major mode switch is turned ON:



```
#

        PRINT LISTINGS (COMMANDS: PY, PN, PO, PA)                PREPARE MODE
        --------------
LISTING STATUS  DESCRIPTION
   N      ON    NETWORK DESCRIPTION (.SR)
   X      ON    CREATE AND PRINT NETWORK CONNECTION CROSS-REFERENCES (.XR)
   W      ON    CREATE AND PRINT WARNINGS, UNUSED PINS, AND STATISTICS (.WN)
   T      ON    TEST PROGRAM (.TP)
   E      ON    MACRO-EXPANDED TEST PROGRAM (.ET)
   A      ON    AUTO-PREPARED TEST PROGRAM (.DA)
   V      ON    DIFFERENCES BETWEEN ORIGINAL AND AUTO-PROG TEST PROGRAMS (.DF)
   G      ON    GOOD BOARD SIMULATION  VALUES (PRETTY PRINTOUT)
   P      ON    PIN ACTIVITY - FIRST 0 AND 1 STATE ACTIVATION STEP NUMBERS
   J      ON    JUMP POINT AND START POINT FILES (.US, .UJ, & .UD)
   F      ON    FAULT CLASSES MODELED FOR DETECTION SIMULATION (.FP)
   D      ON    DETECTED FAULT CLASSES AND PERCENTAGE DETECTION (.FG)
   SF     ON    SELECTED FAULT CLASSES MODELED FOR DETECTION SIMULATION (.SP)
TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

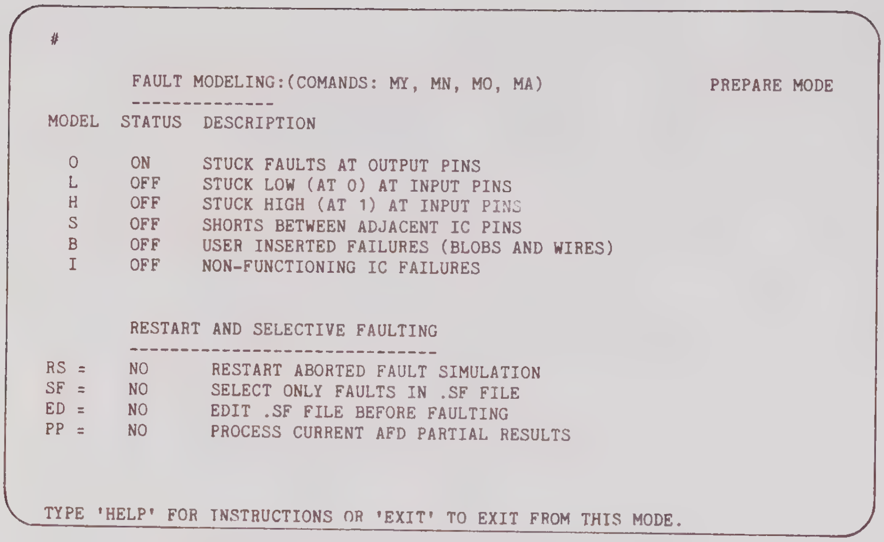
Figure 5-16. The Prepare Mode PRINT LISTING Page.

Listing Switch ON	Listing	Sub Mode Switch ON
N	Network Description (.SR file)	N
X	Network Connection Cross-Ref (.XR File)	
W	Warnings & Statistics (.WN file)	
T	Test Program (.TP file)	T
E	Macro Expanded Test Program (.ET file)	
A	Digitally Auto-Prog TP (.DA file)	G
V	Differences Between .DA & .TP (.DF file)	
G	Good Board Simulation Values (PRETTY)	
P	Pin Activity (ACTIVE)	
J	Start Pt (.US), Jump Pt (.UJ), Pulse Discrimination (.UD) and Conditional Connectivity (.UC)	P
F	Fault Classes Modeled (.FP file)	F
D	Detected Fault Classes & % Det (.FG file)	
SF	Selected Fault Classes Modeled (.SP file)	

5.3.5 Fault Modeling Options

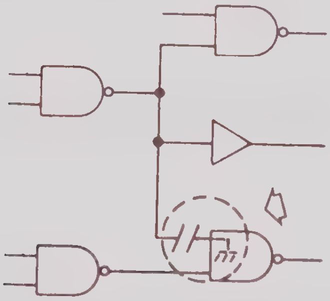
The FAULT MODELING page is the fourth page that can be displayed. The page contains a series of letter switches (O,L,H,S,B,I) that allow individual selection of the following faults when the sub-mode F switch is ON (see Figure 5-17):

Model Switch ON	Faults Modeled
O	Models stuck faults at IC Outputs only. Both SA0 and SA1 faults are modeled. This is a fast method of fault simulation because it does not require that Input faults be simulated and yet does provide a rapid indication of how well the test program can exercise a board. It simulates those failures that correspond to an entire signal line being SA0 and SA1. This type of failure could be physically simulated on an actual board by tying each IC Output pin alternately to V_{cc} and then to GND (although such a procedure could damage IC's). It does provide a rapid indication of how effective the test program is in exercising a board (clocking data through a board, enabling data, etc.) because each IC Output pin of the good model must change state in order for this type of failure to be detected.
L	Models SA0 faults at IC Inputs only. This type of failure is not very likely to occur in TTL or DTL circuits as it requires the dual set of conditions shown in Figure 5-18.
H	Models SA1 faults at IC Inputs only. This type of fault modeling is very useful in detecting breaks that appear to be SA1 in TTL and DTL circuits.
S	Models shorts between adjacent IC pins. This type of failure may occur frequently in a production environment.



```
#

       FAULT MODELING:(COMANDS: MY, MN, MO, MA)              PREPARE MODE
       --------------
MODEL  STATUS  DESCRIPTION

  O     ON     STUCK FAULTS AT OUTPUT PINS
  L     OFF    STUCK LOW (AT 0) AT INPUT PINS
  H     OFF    STUCK HIGH (AT 1) AT INPUT PINS
  S     OFF    SHORTS BETWEEN ADJACENT IC PINS
  B     OFF    USER INSERTED FAILURES (BLOBS AND WIRES)
  I     OFF    NON-FUNCTIONING IC FAILURES


       RESTART AND SELECTIVE FAULTING
       ------------------------------
RS =    NO      RESTART ABORTED FAULT SIMULATION
SF =    NO      SELECT ONLY FAULTS IN .SF FILE
ED =    NO      EDIT .SF FILE BEFORE FAULTING
PP =    NO      PROCESS CURRENT AFD PARTIAL RESULTS


TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-17. The Prepare Mode FAULT MODELING Page.



Conditions Needed

1. Drive Signal Line Broken.
2. Device Internally Shorted.

Figure 5-18. IC Inputs SAO.

B	Models user inserted failures (BLOBS and WIRES). A BLOB simulates a solder blob between two adjacent tracks on a circuit board, while a WIRE simulates plated-through holes or weak points in wires. Either fault may be modeled if B is ON and the BLOB and/or WIRE has been coded into the network description (.SR file) using the procedures in Section 2.
I	Models non-functioning IC failures such as loss of IC power (V_{CC}) or any other malfunction that would completely disable the IC (the IC is <u>dead</u>).

Any combination of faults can be modeled by turning ON the correct model switches. Judicious selection of the available fault modeling options is important because it may have an effect on the length of time it takes to complete a simulation run and may also affect the diagnostic accuracy of the AFL Module during UUT testing in the DIAGNOSE Mode. Only those faults that have been modeled during a PREPARE Mode run can be detected by AFL in the DIAGNOSE Mode.

5.3.6 Restart and Selective Faulting Options

These options are also located on the FAULT MODELING page shown in Figure 5-17, and perform the following unique operations:

1. Restart Option (RS=).

It is possible to stop (abort) the AFD fault simulation during a run, and then re-start the run again at the point of termination. During the AFD run, a BUSY message is displayed that includes the FAULT SIMULATION STATUS for the board being run (number of fault classes simulated, detected, etc.). The message is continually updated as every six fault classes are modeled. The last message entry reads:

LAST CHECK POINT TAKEN AT FAULT CLASS = 288

If the Teletype CTRL and X keys are depressed during a run (regardless of whether the re-start option is ON or OFF), the AFD simulation will be *terminated* after completing the fault class listed in the message (in this case 288), and the Monitor will be entered. In addition, a Teletype message of the form

AFD WILL RESTART ON (BOARD) WITH 288 FAULT CLASSES DONE

will occur. When another AFD run is started on this board it will *begin* with the *next* fault class to be simulated (in this case 289) *if* the run is initiated with the re-start option ON (RS=YES).

The RESTART Option may also be used if the fault simulation run is terminated for *any* reason (use of CTRL C to enter the CAPS or APG Monitors, o loss of test system power, etc.). Under these less than ideal conditions, the AFD simulation will begin again at or near the point of termination, however, some fault classes may not be modeled during the re-run.

2. Selective Faulting (SF=,ED=).

Selective Fault Simulation can be used to *re-run* selected faults *after* a fault simulation run has been completed. This feature makes it possible to determine if a modified test program will now detect failures missed by the original test program in a fraction of the time required for a complete run. The faults that are to be selectively run must be included in an .SF file. The file can be prepared by EDIT using the procedures contained in para 5.2.6.3. If the ED= option is also on (ED=YES), EDIT will be entered automatically prior to the start of an AFD run so that the .SF file can be created. After creating the file, the X and XMIT keys can be depressed to start the AFD run for selective faults. If any changes have been inserted into the *middle* of a test program, it will be necessary to again complete a normal AFD run (with Selective Faulting turned OFF) after completing the selective fault run. If changes have been *appended* to the end of the program, it will not be necessary to re-run the fault simulation again.

3. Process Partial AFD Results (PP=).

During a normal Fault Detection Simulation run with the Process Partial AFD Results Option *off* (PP=NO), detectable failures are identified and collapsed into fault dictionary files that are required by AFL and the AFL to Probe Link for effective fault analysis in the Diagnosis Mode. Next, listing files such as the Fault Class Sets Detected & % Detection (.FG) file and Fault Classes Modeled (.FP) file are output. These files are important when evaluating how effective the test program is at exercising a board, but are not normally available until after the AFD run has been completed. Partial (not complete) AFD results can be processed, however, to obtain output listings of sufficient quality for test program analysis without completing a complete AFD run, if desired. The procedure is as follows:

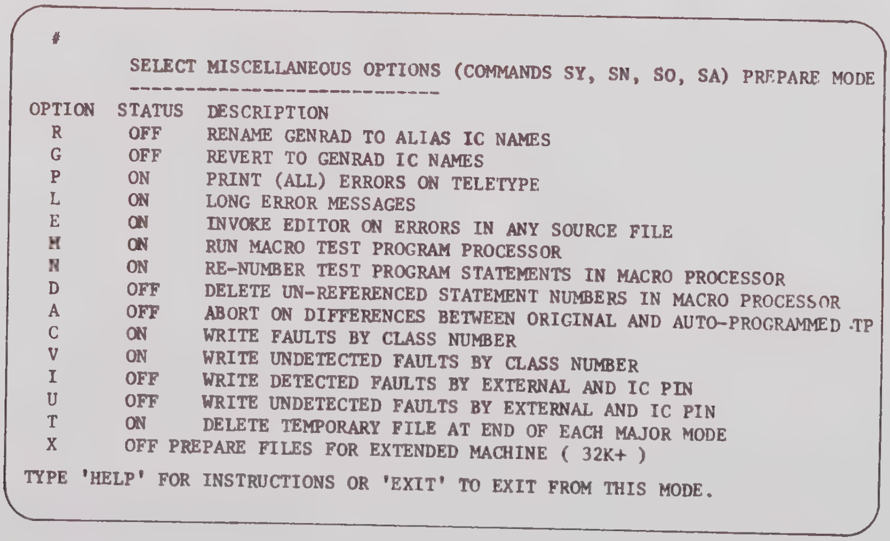
a. Start the initial Fault Detection Simulation Run (or Selected Fault Simulation Run) with the Process Partial AFD Results Option off (PP=NO).

b. Prior to completing the AFD Fault Simulation run, depress the Teletype CTRL and X keys to terminate the AFD Simulation and re-enter the Monitor.

c. Next, re-enter the PREPARE Mode and turn the Partial AFD Results Option on (PP=YES) before starting the run again.

d. When the run is initiated with PP=YES, the AFD Module will be by-passed and the .FG and .FP files output to the listing device specified. Note that information in the files will reflect the percentage of detection completed when the last AFD run was terminated by a CTRL X command. Sufficient data may be available, however, for test program analysis.

e. After using the Partial AFD Results Option and prior to entering the Diagnosis Mode for board testing, always re-run the F Mode again with the Partial AFD Results Option off (PP=NO) so that a complete set of fault dictionary files will be available for testing purposes.



```
 #
         SELECT MISCELLANEOUS OPTIONS (COMMANDS SY, SN, SO, SA) PREPARE MODE
         ----------------------------
OPTION  STATUS  DESCRIPTION
  R      OFF    RENAME GENRAD TO ALIAS IC NAMES
  G      OFF    REVERT TO GENRAD IC NAMES
  P      ON     PRINT (ALL) ERRORS ON TELETYPE
  L      ON     LONG ERROR MESSAGES
  E      ON     INVOKE EDITOR ON ERRORS IN ANY SOURCE FILE
  M      ON     RUN MACRO TEST PROGRAM PROCESSOR
  N      ON     RE-NUMBER TEST PROGRAM STATEMENTS IN MACRO PROCESSOR
  D      OFF    DELETE UN-REFERENCED STATEMENT NUMBERS IN MACRO PROCESSOR
  A      OFF    ABORT ON DIFFERENCES BETWEEN ORIGINAL AND AUTO-PROGRAMMED .TP
  C      ON     WRITE FAULTS BY CLASS NUMBER
  V      ON     WRITE UNDETECTED FAULTS BY CLASS NUMBER
  I      OFF    WRITE DETECTED FAULTS BY EXTERNAL AND IC PIN
  U      OFF    WRITE UNDETECTED FAULTS BY EXTERNAL AND IC PIN
  T      ON     DELETE TEMPORARY FILE AT END OF EACH MAJOR MODE
  X      OFF PREPARE FILES FOR EXTENDED MACHINE ( 32K+ )
TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-19. The Prepare Mode SELECT MISCELLANEOUS OPTIONS Page.

5.3.7 Select Miscellaneous Options

The SELECT MISCELLANEOUS OPTIONS page is the fifth page that can be displayed. The page contains a series of special purpose letter options that may be used with various sub mode options (see Figure 5-19). The following combinations are possible:

Misc. Option ON	Description	Major Mode Option ON
P	Print (all) errors on Teletype	N
L	Long Error Messages	
E	Invoke EDIT on errors	
T	Delete Temp Files at end of run	
R	Rename GenRad to Alias IC Names	
G	Revert to GenRad IC Names	
X	Prepare Files for Extended Machine (32K +)	
P	Print (all) errors on Teletype	T
L	Long Error Messages	
E	Invoke EDIT on errors	
M	Run Macro TP Processor	
N	Re-number TP statements in Macro	
D	Delete Unref. Statements in Macro	
T	Delete Temp Files at end of run	
P	Print (all) errors on Teletype	G
A	Abort on Diff between .DA and .TP	
T	Delete Temp Files at end of run	
T	Delete Temp Files at end of run	D
P	Print (all) errors on Teletype	F
C	Print Faults in .FP file by Class No.	
V	Print Undetected Faults in .FP file by Class No.	
I	Print Detected Faults in .FP file by EXT & IC Pin	
U	Print Undetected Faults in .FP file by EXT & IC Pin	
T	Delete Temp Files at end of run	
E	Invoke EDIT on errors	U,T, P,Z
T	Delete Temp Files at end of run	
R	Rename GenRad to Alias IC Names	
G	Revert to GenRad IC Names	
P	Print (all) errors on Teletype	I
L	Long Error Messages	
E	Invoke EDIT on errors	

Each of the Miscellaneous options listed are described in the following paragraphs.

5.3.7.1 Print (All) Errors on Teletype (P).

Normally, error messages are output to the Teletype or to the listing device specified by the FILE USAGE Page LI= Option (depending on the software module in use when the error occurs). If the P Option is ON, all error messages will be output to the Teletype, and will not be printed on the listing device specified by the LI= Option.

5.3.7.2 Long Error Messages (L).

Long error messages define errors in clear text (L ON). When L is OFF, abbreviated error messages are generated. Some examples of long and short error messages are provided in Para. 5.2.2.2.

5.3.7.3 Invoke EDIT on Errors (E).

This option allows EDIT to be entered if errors are detected in the source file being processed by CAPS during a PREPARE Mode run. If E is not active and errors occur, error messages will be generated, execution will be suppressed, and the Monitor will be entered. When E is active, EDIT is entered automatically so that the source file being processed can be corrected "on-line". Once corrections have been made, and the corrected file is in core, depressing the EDIT Mode X and XMIT keys will result in an immediate return to the aborted CAPS run, and the run will begin again at or near the point of termination. If EDIT is exited via the E and XMIT command, a return to the Monitor will take place.

5.3.7.4 Run Macro Test Program Processor (M).

If the M option is ON, macro commands in the basic test program (.TP file) will be expanded by a macro processor. The output of the macro processor is an expanded test program (.ET file). Some examples of expanded macro commands are provided in Para. 5.2.3.1. The .ET file is then translated by the CAPS Translator (CTRAN) to obtain the binary data base required by CAPS or APG to complete other major mode runs.

5.3.7.5 Re-Number Test Program Macro Statements (N).

If the N option is also ON (with M), all labelled commands in the basic test program .TP file will be re-numbered in the expanded .ET file. The re-numbering sequence starts with 1 and goes up in unbroken sequence (1,2,3,4,etc.). Details are contained in Para. 5.2.3.1.

5.3.7.6 Delete Unreferenced TP Statements (D).

If the D option is also ON (with M), unreferenced labels (labels not referenced by CALLs, etc.) will be deleted from the expanded test program as the macro processor is run. It is thus possible to add labels to aid in test program debugging, and then delete the labels later in order to speed up program execution in the DIAGNOSE Mode.

5.3.7.7 Abort on Differences Between .TP & .DA Files (A).

This option should be used if the basic test program .TP file contains Output (OH, OL) statements. While Output statements are not required by CAPS or APG, they may be included in the test program. When A is ON, CAPS or APG compare the simulated Outputs that have been created and inserted into the digitally auto-programmed test program (.DA file) with user inserted Outputs from the basic .TP file and flag any differences.

5.3.7.8 Write Faults By Class Number (C).

When the C option is ON, the .FP file will contain a FAULT DETECTION BY CLASS NO. listing that includes all detected and undetected faults arranged in numerical order (1,2,3,4,etc.) by class. The first test step at which the class was detected and any equivalent faults are also listed.

5.3.7.9 Write Undetected Faults By Class Number (V).

When the V option is ON, the .FP file will also include a listing of UNDETECTED FAULTS BY CLASS NO. This listing can be suppressed by turning V OFF.

5.3.7.10 Write Detected Faults By EXT & IC Pin (I).

When the I option is ON, the .FP file will also include a listing of EXTERNAL FAULT DETECTION by EXT and IC pin. This listing can be suppressed by turning I OFF.

5.3.7.11 Write Undetected Faults By EXT & IC Pin (U).

When the U option is ON, the .FP file will also include a listing of UNDETECTED EXTERNAL FAULTS by EXT and IC pin. This listing can be suppressed by turning U OFF.

5.3.7.12 Delete Temporary Files (T).

When T is ON, CAPS temporary files, created during each sub-mode run, but no longer required, will be deleted. Thus, files that are no longer required are not saved. It is recommended that this option be turned ON (T ON) when using the PREPARE Mode since the temporary files that would otherwise be saved are of no use to the user and will only tend to consume space on the disk(s).

5.3.7.13 Prepare Files for Extended Machine, 32K + (X).

The X option enables the building of large simulation files. This option builds a CAPS Simulation Table (.ST file) for 64K (when ON) or for 32K (when OFF) as the N Sub-mode of PREPARE is run. The .ST file is then used by the other Sub-modes of PREPARE as they are run. It is therefore important to turn the X option ON or OFF prior to running the N Sub-mode in order to build the correct .ST data base file.

5.3.7.14 Rename Alias ICs (R) and Revert to Original Names (G).

The R option enables the rename mode, and the G option enables the revert mode of the ALIAS.SV module. Refer to para 5.2.2.3 for details.

5.3.8 Display Options

The DISPLAY OPTIONS page is the sixth PREPARE Mode page to be displayed (see Figure 5-20). The page allows user configuration of the good board simulation results (PRETTY) and pin activation results (ACTIVE) that are available as a result of a successful G sub-mode run. Options on this page affect the - PRETTY and ACTIVE displays that are output at the time of a good board simulation (sub-mode G run). It is also possible to use the Utility Program DISPLAY to output complete or selected portions of PRETTY once a G Mode run has been completed. The R, S, N, I, and O options allow a user to specify the amount of data to be included in PRETTY and ACTIVE in the following manner:

Option ON	Displays
R	Lists the simulation results by EXTERNAL INPUTS also.
S	Lists the simulation results by EXTERNAL OUTPUTS also.
N	Lists the simulation results by INTERMEDIATE names also.
I	Lists the simulation results by IC INPUTS also.
O	Lists the simulation results by IC OUTPUTS also.

The last option (D) changes the format of the PRETTY file by listing unchanged 0, 1, and X states as:

(0 . . . . 1 , , , , X - - - -)

If D is OFF, unchanged states will be listed as:

(0 0 0 0 0 1 1 1 1 1 X X X X X)

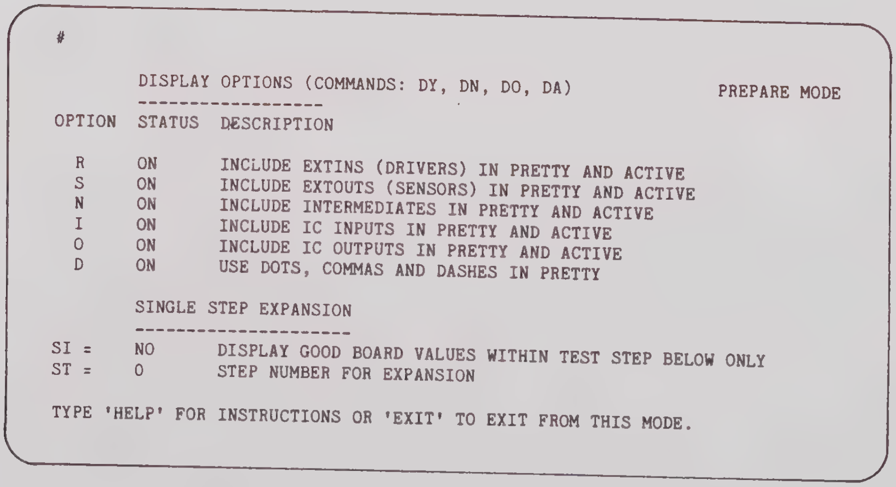
5.3.9 Single Step Expansion

Single step expansion is useful when it is desired to expand active test steps in order to observe the number of transitions that actually occur. An expanded run is initiated in the following manner:

1. Turn the SI Option ON (SI=YES) and specify the simulation step number from the Good Board Simulation Results (PRETTY) file that contains the state to be expanded (ST=20).

2. Next, perform a G Mode run with the SI option turned ON to build the expanded CAPS test data.

3. Next, use the Utility Program DISPLAY to output the expanded test step data.



```
#

        DISPLAY OPTIONS (COMMANDS: DY, DN, DO, DA)              PREPARE MODE
        ------------------
OPTION  STATUS  DESCRIPTION

  R     ON      INCLUDE EXTINS (DRIVERS) IN PRETTY AND ACTIVE
  S     ON      INCLUDE EXTOUTS (SENSORS) IN PRETTY AND ACTIVE
  N     ON      INCLUDE INTERMEDIATES IN PRETTY AND ACTIVE
  I     ON      INCLUDE IC INPUTS IN PRETTY AND ACTIVE
  O     ON      INCLUDE IC OUTPUTS IN PRETTY AND ACTIVE
  D     ON      USE DOTS, COMMAS AND DASHES IN PRETTY

        SINGLE STEP EXPANSION
        ---------------------
SI =    NO      DISPLAY GOOD BOARD VALUES WITHIN TEST STEP BELOW ONLY
ST =    0       STEP NUMBER FOR EXPANSION

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-20. The Prepare Mode DISPLAY OPTIONS Page.

5.3.10 Select Inc-Sim Options (APG Only)

Select Inc-Sim Options page is the seventh and last PREPARE Mode page to be displayed. The page contains options that allow incremental simulation to be run when the EXECUTE SUB MODE I Option is ON. This page is only available when using the APG Monitor. Options on the page are described in Appendix B.

5.4 PREPARE MODE DATA OUTPUTS

5.4.1 General

As the N, T, G, F, and P Sub-Modes are run, a series of data outputs are generated that may be output to a listing device (TEC;,LPT:,TTY:), to a disk surface (such as DSK:), or suppressed entirely, if not required. The data formats used are described in the following paragraphs. Partial data outputs are shown in some of the illustrations that follow. Complete listings can be generated at the time of a run or through use of the Utility Programs LISTALL and DISPLAY after completing a run. Data outputs associated with the APG I Sub-Mode are described in Appendix B.

5.4.2 N Sub-Mode Data Outputs

The data outputs listed in the following paragraphs can occur as a result of a successful N Sub-Mode run (Network Description .SR Analysis). Each output can be suppressed by turning the associated PRINT LISTINGS option OFF.

5.4.2.1 Input Listing (.SR file)

The Input Listing is a complete printout of the .SR file that has been run through the Pre-Processor (see Figure 5-21). Line numbers have been added to the listing by CAPS or APG and correspond to pre-processor error message line numbers. The listing can be suppressed by turning the PRINT LISTINGS N Option OFF.

5.4.2.2 Warnings & Statistics (.WN File)

Warnings and statistics for the .SR file are output next (see Figure 5-19). The Warnings section includes a complete list of those ICs that have unused IC Input pins, Output pins, mis-matches, or overloads. The statistics section lists statistical data that may be of value to a user (number of external Inputs, Outputs, busses, network nodes, etc.). The CORE ALLOCATION data is of particular importance because it reflects the total amount of core that will be required to complete a fault simulation run.

IC mis-matches (CMOS driving TTL for example) and IC overloads are identified further in the cross-reference (.XR file) listing.

Wired-AND statistics are also listed . In the sample board shown, six wired-AND circuits exist, and each of the circuits has two <u>sources</u> (listed under SIZE) that are driving it.

The Warnings and Statistics listing can be suppressed by turning the PRINT LISTINGS W Option OFF.

The CORE ALLOCATION data shown in Figure 5-22 can be analyzed in the following manner:

```
CORE ALLOCATION:
                    10000 - 14567
                    23000 - 23153
```

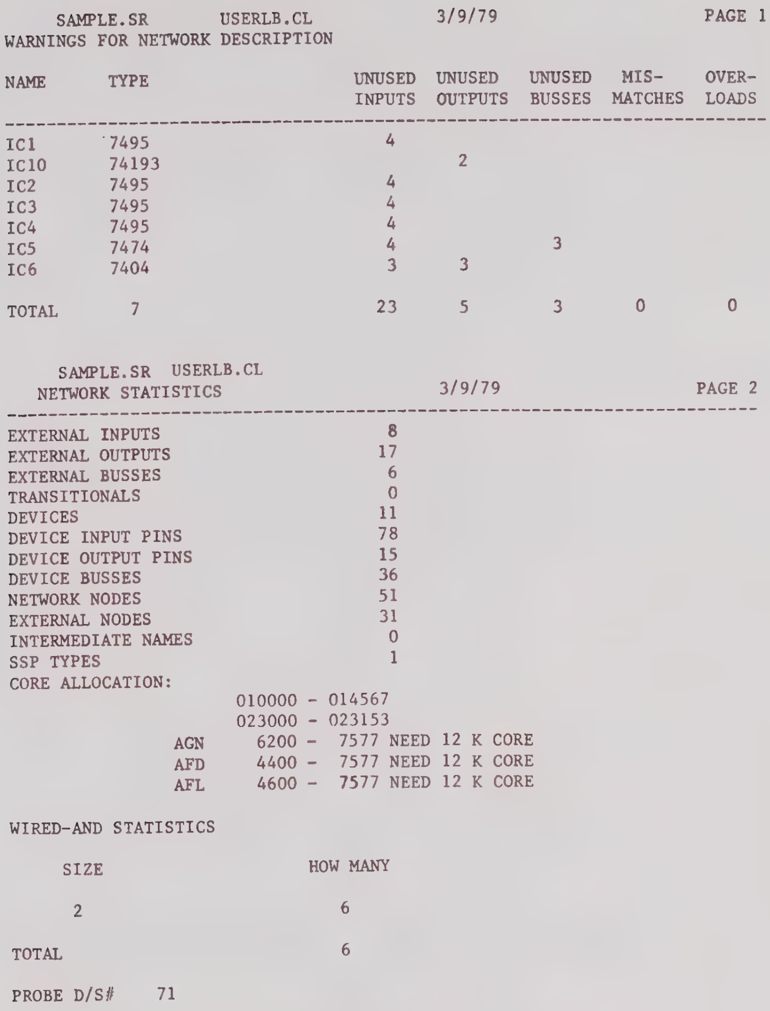
```
****  SAMPLE.SR  PROCESSED BY CAPS, VERSION 46  U= USERLB.CL ON 6/14/1979  ****
0001:   J1PIN76 EXTIN   76,IC5.4 SETFF;
0002:   J1PIN6  EXTOUT  6,IC5.5 QOUT;
0003:   J1PIN5  EXTIN   5,IC6.1 CLRFF;
0004:   J1PIN7  EXTIN   7,IC1.9 IC6.5 IC2.9 IC3.9 IC4.9 RC;
0005:   J1PIN8  EXTIN   8,IC1.1 SI;
0006:   J1PIN25 EXTIN   25,IC7.14 IC8.14 IC9.14 IC10.14 CLRCNT;
0007:   J1PIN26 EXTIN   26,IC7.11 IC8.11 IC9.11 IC10.11 LD;
0008:   J1PIN27 EXTIN   27,IC7.5 CUP1;
0009:   J1PIN28 EXTIN   28,IC7.4 CDN1;
0010:   J1PIN29 EXT     29,IC7.12 IC8.5 CUP2;
0011:   J1PIN30 EXT     30,IC7.13 IC8.4 CDN2;
0012:   J1PIN31 EXT     31,IC8.12 IC9.5 CUP3;
0013:   J1PIN32 EXT     32,IC8.13 IC9.4 CDN3;
0014:   J1PIN33 EXT     33,IC9.12 IC10.5 CUP4;
0015:   J1PIN34 EXT     34,IC9.13 IC10.4 CDN4;
0016:   J1PIN35 EXTOUT  35,IC7.2 QA1;
0017:   J1PIN36 EXTOUT  36,IC7.3 QB1;
0018:   J1PIN37 EXTOUT  37,IC7.6 QC1;
0019:   J1PIN38 EXTOUT  38,IC7.7 QD1;
0020:   J1PIN39 EXTOUT  39,IC8.2 QA2;
0021:   J1PIN40 EXTOUT  40,IC8.3 QB2;
0022:   J1PIN41 EXTOUT  41,IC8.6 QC2;
0023:   J1PIN42 EXTOUT  42,IC8.7 QD2;
0024:   J1PIN43 EXTOUT  43,IC9.2 QA3;
0025:   J1PIN44 EXTOUT  44,IC9.3 QB3;
0026:   J1PIN45 EXTOUT  45,IC9.6 QC3;
0027:   J1PIN46 EXTOUT  46,IC9.7 QD3;
0028:   J1PIN47 EXTOUT  47,IC10.2 QA4;
0029:   J1PIN48 EXTOUT  48,IC10.3 QB4;
0030:   J1PIN49 EXTOUT  49,IC10.6 QC4;
0031:   J1PIN50 EXTOUT  50,IC10.7 QD4;
0032:   IC1     7495    J1PIN8,6=GND,8=GND,J1PIN7,IC2.1 IC7.9,IC7.10,
0033:                   IC7.1,IC7.15;
0034:   IC2     7495    IC1.10,6=GND,8=GND,J1PIN7,IC3.1 IC8.9,IC8.10,
0035:                           IC8.1,IC8.15;
0036:   IC3     7495    IC2.10,6=GND,8=GND,J1PIN7,IC4.1 IC9.9,IC9.10,
0037:                   IC9.1,IC9.15;
0038:   IC4     7495    IC3.10,6=GND,8=GND,J1PIN7,IC5.2 IC10.9,IC10.10,
0039:                   IC10.1,IC10.15;
0040:   IC5     7474    IC6.4,IC4.10,IC6.6,J1PIN76,J1PIN6;
0041:   IC6     7404    J1PIN5,IC6.3,IC6.2,IC5.1,J1PIN7,IC5.3;
0042:   IC7     74193   IC1.12,J1PIN35,J1PIN36,J1PIN28,J1PIN27,J1PIN37,
0043:                   J1PIN38,9=IC1.10,IC1.11,J1PIN26,J1PIN29,J1PIN30,
0044:                   J1PIN25,IC1.13;
0045:   IC8     74193   IC2.12,J1PIN39,J1PIN40,J1PIN30,J1PIN29,J1PIN41,
0046:                   J1PIN42,9=IC2.10,IC2.11,J1PIN26,J1PIN31,J1PIN32,
0047:                   J1PIN25,IC2.13;
0048:   IC9     74193   IC3.12,J1PIN43,J1PIN44,J1PIN32,J1PIN31,J1PIN45,
0049:                   J1PIN46,9=IC3.10,IC3.11,J1PIN26,J1PIN33,J1PIN34,
0050:                   J1PIN25,IC3.13;
0051:   IC10    74193   IC4.12,J1PIN47,J1PIN48,J1PIN34,J1PIN33,J1PIN49,
0052:                   J1PIN50,9=IC4.10,IC4.11,J1PIN26,14=J1PIN25,
0053:                           IC4.13;
0054:   #PROBE=71;
0055:           %
0056:           IC10.9=NOUV;
```

Figure 5-21. Input Listing (.SR File) Format.

The CAPS/APG system code requires up to 4567_8 words in field 1, and the area between 3000 - 3153_8 in field 2 for the run. At least 12k of core is therefore required in order to make this run (fields 0, 1, 2 are needed).

AGN   6200 - 7577   NEED 12K CORE

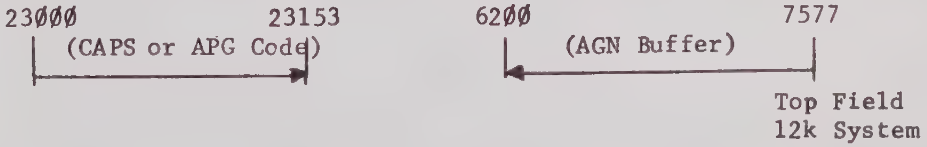
A buffer for the AGN (Analysis of Good Network) module is required for a G Mode run. The buffer uses the area between 6200 - 7577_8 in the top field of core for this run. Since the buffer will always begin in the top field of core, the field is not listed. No overlapping is permitted between the area



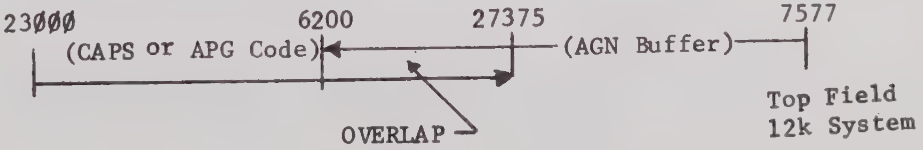
```
     SAMPLE.SR      USERLB.CL            3/9/79                    PAGE 1
WARNINGS FOR NETWORK DESCRIPTION

NAME       TYPE                       UNUSED  UNUSED   UNUSED   MIS-     OVER-
                                      INPUTS  OUTPUTS  BUSSES   MATCHES  LOADS
------------------------------------------------------------------------------
IC1        7495                          4
IC10       74193                                2
IC2        7495                          4
IC3        7495                          4
IC4        7495                          4
IC5        7474                          4                3
IC6        7404                          3      3

TOTAL        7                          23      5         3        0        0

     SAMPLE.SR  USERLB.CL
   NETWORK STATISTICS                    3/9/79                    PAGE 2
-----------------------------------------------------------------------------
EXTERNAL INPUTS                          8
EXTERNAL OUTPUTS                        17
EXTERNAL BUSSES                          6
TRANSITIONALS                            0
DEVICES                                 11
DEVICE INPUT PINS                       78
DEVICE OUTPUT PINS                      15
DEVICE BUSSES                           36
NETWORK NODES                           51
EXTERNAL NODES                          31
INTERMEDIATE NAMES                       0
SSP TYPES                                1
CORE ALLOCATION:
                       010000 - 014567
                       023000 - 023153
                 AGN     6200 -   7577 NEED 12 K CORE
                 AFD     4400 -   7577 NEED 12 K CORE
                 AFL     4600 -   7577 NEED 12 K CORE

WIRED-AND STATISTICS

     SIZE                          HOW MANY

       2                              6

TOTAL                                 6

PROBE D/S#    71
```

Figure 5-22. Warnings and Statistics (.WN File).

assigned for CAPS or APG system code and the area assigned for each buffer. In the following example, enough core is available for an AGN run without any overlapping.


23000
23153
(CAPS or APG Code)
6200
7577
(AGN Buffer)
Top Field
12k System

If the CORE ALLOCATION requirements had been 23000 - 27375, then a G Mode run could not have been performed successfully in 12k as insufficient AGN buffer space would have been available as shown in the following example:


23000
6200
27375
7577
(CAPS or APG Code)
(AGN Buffer)
OVERLAP
Top Field
12k System

Note that the CAPS or APG system code and AGN buffer requirements have overlapped, thus making it impossible to perform a successful G Mode run with a 12k test system. If a test system with 16k of core (or more) were being used, the run could have been performed. In addition, the buffer area listed (AGN 6200 - 7577) must be shifted down by 200 words in those test systems that are only equipped with 12k of core. For example, a user with 12k of core would subtract 200 words from both buffer limits (AGN 6000 - 7377) before determining whether any overlap has occurred.

A buffer for the AFD (Automatic Fault Detection) module is also required for an F Mode run. The buffer uses the area between 4400 - 7577_8 in the top field of core for the run. As with the AGN buffer, no overlapping of code is permitted and 200 words must be subtracted from the buffer totals in 12k test systems before determining whether any overlap has occurred.

A buffer for the AFL (Automatic Fault Location) module is also required for a DIAGNOSE Mode run with AFL. The buffer uses the area between 2400 - 7577_8 in the top field of core for the run. As with the AGN buffer, no overlapping of code is permitted and 200 words must be subtracted from the buffer totals in 12k test systems before determining whether any overlap has occurred.

NOTES

- If sufficient core is not available, the N Sub-mode can be re-run with the SELECT MISC Page X Option turned ON. The G and F Sub-modes can be run only if a 64K test system is available. The N, T, D, Q, P, and Z Sub-modes can be run on 32K test systems, if desired (refer to Para. 5.3.7.7).
- AFL will do an initial AFL lookup in the DIAGNOSE Mode and then branch to the guided probe without completing an AFL Simulation whenever 32K of memory is not sufficient.

5.4.2.3 Network Connection Cross-Reference Data (.XR File)

Network connection cross-reference data is output next (see Figure 5-23). This is a large file (.XR file) that includes a complete listing of circuit INput, OUTput, and BUSS connections by assigned External NAME, TYPE, Driver/Sensor (D/S) pin number, external shelf pin connections, loading, and CONNECTIONS (where the EXTERNAL NAME is connected to). The file is listed both by EXTERNAL NAME connections and by IC CONNECTIONS (cross-referenced). Asterisks (*) in the CONNECTIONS column indicate any signal sources that are part of the node. Blank brackets [ ] appear before each complete node the first time it is listed. Check marks can be placed in the brackets [✓] to serve many purposes (perhaps to identify those circuits that have been wire-wrapped, etc.). Each CONNECTIONS list also includes the .SR file Line Number at which the pin was first connected, and logic family associated with that node (LN=12,FAM=TTL). External test system D/S shelf pin connections (J21-50) are also listed. The last page of the listing includes any wired-AND connections that may occur. NOPROBE, NOFAULT, or other similar points are automatically identified in the listing.

The cross-reference list can be suppressed during a run by turning the PRINT LISTINGS X Option OFF or by listing DSK: as the Output Device. If a run is performed with the X Option OFF, the .XR file will not be created. However, if LISTALL Mode is called asking for the .XR file, XREF will be run to create it.

```
    SAMPLE.SR      USERLB.CL           6/14/79             PAGE   1
NAME   TYPE D/S#          LOAD      CONNECTIONS          EXTERNALS
-----------------------------------------------------------------------
J1PIN25  IN     25 J21-50  4.00 [ ] CLRCNT IC10.14 IC9.14 IC8.14 IC7.14
                                  (LN=6)
J1PIN26  IN     26 J21-49  4.00 [ ] LD IC10.11 IC9.11 IC8.11 IC7.11
                                  (LN=7)
J1PIN27  IN     27 J21-48  1.00 [ ] CUP1 IC7.5 (LN=8)
J1PIN28  IN     28 J21-47  1.00 [ ] CDN1 IC7.4 (LN=9)
J1PIN29  BUSS   29 J21-46  1.00 [ ] IC7.12* CUP2 IC8.5 (LN=10)
J1PIN30  BUSS   30 J21-45  1.00 [ ] IC7.13* CDN2 IC8.4 (LN=11)
J1PIN31  BUSS   31 J21-44  1.00 [ ] IC8.12* CUP3 IC9.5 (LN=12)
J1PIN32  BUSS   32 J21-43  1.00 [ ] IC8.13* CDN3 IC9.4 (LN=13)
J1PIN33  BUSS   33 J21-42  1.00 [ ] IC9.12* CUP4 IC10.5 (LN=14)
J1PIN34  BUSS   34 J21-41  1.00 [ ] IC9.13* CDN4 IC10.4 (LN=15)
J1PIN35  OUT    35 J21-40       [ ] QA1 IC7.2* (LN=16)
J1PIN36  OUT    36 J21-39       [ ] QB1 IC7.3* (LN=17)
J1PIN37  OUT    37 J21-37       [ ] QC1 IC7.6* (LN=18)
J1PIN38  OUT    38 J21-36       [ ] QD1 IC7.7* (LN=19)
J1PIN39  OUT    39 J21-35       [ ] QA2 IC8.2* (LN=20)
J1PIN40  OUT    40 J21-34       [ ] QB2 IC8.3* (LN=21)
J1PIN41  OUT    41 J21-33       [ ] QC2 IC8.6* (LN=22)
J1PIN42  OUT    42 J21-32       [ ] QD2 IC8.7* (LN=23)
J1PIN43  OUT    43 J21-31       [ ] QA3 IC9.2* (LN=24)
J1PIN44  OUT    44 J21-30       [ ] QB3 IC9.3* (LN=25)
J1PIN45  OUT    45 J21-29       [ ] QC3 IC9.6* (LN=26)
J1PIN46  OUT    46 J21-28       [ ] QD3 IC9.7* (LN=27)
J1PIN47  OUT    47 J21-27       [ ] QA4 IC10.2* (LN=28)
J1PIN48  OUT    48 J21-26       [ ] QB4 IC10.3* (LN=29)
J1PIN49  OUT    49 J22-1        [ ] QC4 IC10.6* (LN=30)
J1PIN5   IN      5 J21-5   1.00 [ ] CLRFF IC6.1 (LN=3)
J1PIN50  OUT    50 J22-2        [ ] QD4 IC10.7* (LN=31)
J1PIN6   OUT     6 J21-6        [ ] QOUT IC5.5* (LN=2)
J1PIN7   IN      7 J21-7   5.00 [ ] RC IC4.9 IC3.9 IC2.9 IC6.5 IC1.9
                                  (LN=4)
J1PIN76  IN     76 J22-47  2.00 [ ] SETFF IC5.4 (LN=1)
J1PIN8   IN      8 J21-8   1.00 [ ] SI IC1.1 (LN=5)

    SAMPLE.SR      USERLB.CL           6/14/79             PAGE   2
NAME      PINS     LOAD          CONNECTIONS        DEVICES
-----------------------------------------------------------------------

GND       GND                        (NOPROBE)
          OUT    1               IC4.8 IC4.6 IC3.8 IC3.6 IC2.8 IC2.6 IC1.8
                                 IC1.6 (LN=32,NOFAULTO,NOPRINT)

IC1       7495
          IN     1               J1PIN8* SI (LN=5)
          IN     2             NC (FAM=TTL)
          IN     3             NC (FAM=TTL)
          IN     4             NC (FAM=TTL)
          IN     5             NC (FAM=TTL)
          IN     6          [ ]  GND.1* IC4.8 IC4.6 IC3.8 IC3.6 IC2.8 IC2.6
                                 IC1.8 (LN=32,FAM=TTL)
          GND    7
          IN     8               IC1.6 GND.1* IC4.8 IC4.6 IC3.8 IC3.6 IC2.8
                                 IC2.6 (LN=32,FAM=TTL)
          IN     9               J1PIN7* RC IC4.9 IC3.9 IC2.9 IC6.5
                                    (LN=4)
          BUSS  10  2.00    [ ]  IC7.9 IC2.1 (LN=32,FAM=TTL)
          BUSS  11  1.00    [ ]  IC7.10 (LN=32,FAM=TTL)
          BUSS  12  1.00    [ ]  IC7.1 (LN=33,FAM=TTL)
          BUSS  13  1.00    [ ]  IC7.15 (LN=33,FAM=TTL)
          VCC   14
```

Figure 5-23. Network Connection Cross-Reference (.XR File) Extracts.

5.4.3 T Sub-Mode Data Outputs

The data outputs listed in the following paragraphs can occur as a result of a successful Major Mode T run (Test Program Translation). Each output can be suppressed by turning the associated PRINT LISTINGS option OFF.

5.4.3.1 Basic Test Program Listing (.TP File)

The first data output of a Major Mode T run is the basic test program (.TP file) listing. A typical listing is shown in Figure 5-24. The listing can be suppressed by turning the PRINT LISTINGS T Option OFF.

5.4.3.2 Expanded Test Program (.ET File)

The next output is an expanded test program (.ET file). The first part of the .ET file consists of a list of any alphanumeric pin names used in the network description (.SR file) or a statement to the effect that no alphanumeric pin names were found. The message format is not shown in the .ET file listing that follows, but is described in detail in Para. 5.2.3.1. If test program macro statements are included in the basic test program (.TP file), and the Miscellaneous M Option is ON, the statements will be expaned by a macro processor. The macro processor also checks for syntax and other errors (refer to Para. 5.2.3.1). If the PRINT LISTINGS E Option is also ON, a listing of the expanded test program will occur next (see Figure 5-25). Note that the basic test program macro statements .REPEAT 9; and .LOOP; have been expanded to cover all nine loops (only five are shown in the illustration).

If the Miscellaneous N Option had also been ON, test program statement labels would have been re-numbered in sequence starting with number one (1,2,3,etc.). If the Miscellaneous D Option had also been ON (with N), unreferenced test program labels would have been deleted from the .ET file. It is thus possible to add labels to aid in test program debugging, and then delete the labels later in order to speed up program execution in the DIAGNOSE Mode.

5.4.4 G Sub-Mode Data Outputs

The data outputs listed in the following paragraphs can occur as the result of a successful sub-mode G Run (Good Network Simulation). Each output can be suppressed by turning the associated PRINT LISTINGS Option OFF.

```
BEGIN CAPS 'SAMPLE';
SET PS-1 V=5 I=3;
ASSIGN LGC IA(76,5,7,8,25-28)VILA=0.4 VIHA=3.0
           OA(29-50,71)VOLA=1.0 VOHA=1.5;
IGNORE LGC(71);                 /IGNORE PROBE SENSOR/
10:  IH(26-28,76);              /RESET FF,SET CTR CLOCKS & LOAD/
20:  IL(76) IH(5);              /TEST FF SET/
30:  IH(7,76);                  /SET UP CLOCK/
 .REPEAT 9;
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
 .LOOP;
40: IL(26);                     /LOAD INTO COUNTER/
50: IH(26);
60: IL(7);                      /MOVE CHECKERBOARD OVER/
70: IH(7);
80: IL(26);
90: IH(26);
100:IH(25);                     /CLEAR COUNTER/
110:IL(25);
120:IL(28);
130:IH(28);
140:IL(27);
150:IH(27);
FINISH CAPS;
END;
```

Figure 5-24. The Basic Test Program (.TP File) for the Sample Circuit.

```
/ PIN NAMES FROM SAMPLE.SR     6-14-79 /
 ASSIGN PNAME  J1PIN5=5,J1PIN6=6,J1PIN7=7,J1PIN8=8;



BEGIN CAPS 'SAMPLE';
SET PS-1 V=5 I=3;
ASSIGN LGC IA(76,5,7,8,25-28)VILA=0.4 VIHA=3.0
           OA(29-50,71)VOLA=1.0 VOHA=1.5;
IGNORE LGC(71);                 /IGNORE PROBE SENSOR/
10:  IH(26-28,76);              /RESET FF,SET CTR CLOCKS & LOAD/
20:  IL(76) IH(5);              /TEST FF SET/
30:  IH(7,76);                  /SET UP CLOCK/
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
40: IL(26);                     /LOAD INTO COUNTER/
50: IH(26);
60: IL(7);                      /MOVE CHECKERBOARD OVER/
70: IH(7);
80: IL(26);
90: IH(26);
100:IH(25);                     /CLEAR COUNTER/
110:IL(25);
120:IL(28);
130:IH(28);
140:IL(27);
150:IH(27);
FINISH CAPS;
END;
```

Figure 5-25. The Expanded Test Program (.ET File).

5.4.4.1 Digitally Autoprogrammed Test Program (.DA File)

The digitally autoprogrammed test program (.DA file) is the first output of the good network simulation (see Figure 5-26). During the good network simulation, output patterns are automatically inserted into the test program. In addition, CAPS/APG line numbers / 1 / are automatically added to the .DA file at every ten test steps. The added numbers correspond to step numbers listed in the Simulation Results (PRETTY) file that may be output as G is run. Note that line numbers include all TEST LGC statements only and do not count DO loops, or dummy tests ($) where output checking is suppressed.

The digitally autoprogrammed test program can be suppressed by turning the PRINT LISTINGS A Option OFF.

5.4.4.2 Differences Between .TP & .DA Files (.DF File)

If Output patterns were included in the original test program (.TP file), they will be checked against the simulated Output patterns in the .DA file and any differences that are noted will be listed at this time (if the PRINT LISTINGS V Option is ON).

5.4.4.3 Good Board Simulation Results (PRETTY)

A typical good board simulation results output is shown in Figure 5-27. This is a large output that can be generated during a G Mode run or output after the run through use of the Utility Program DISPLAY. A DISPLAY OPTIONS page can be used to generate the type of PRETTY listing desired. The listing can include EXTERNAL INPUTS, EXTERNAL OUTPUTS, IC INPUTS, and IC OUTPUTS, Intermediate names or any of these categories can be suppressed. In addition, the format used to represent 0, 1, and X states can be altered (refer to Para. 5.3.8).

Symbols used in the listing are identified as follows:

Symbol	Meaning
$	Simulated but not recorded steps have occurred prior to this step (perhaps a DO loop has been included, Output checks have been suppressed, or SET and/or REASSIGN LGC statements have been inserted into the body of the program).
B	A Buss connection.
0	Zero state (Low).
1	One state (High).
X	Unknown state. While an actual board's outputs will be in either a 1 or 0 state, the simulator has decided that because of either critical race conditions, or improper initialization, at this test step, the states cannot be determined.
P	More than one transition has been detected with the signal ending LOW (i.e. 010, 1010, 1010101...0, or 0X0, X0X0, etc.).
N	More than one transition has been detected with the signal ending HIGH (i.e. 101, 0101, 010101...1, or 1X1, X1X1, etc.).
R	A logic hazard exists at this pin. One of the following three events may produce an R. 1. More than one transition between the 1 state and the X state but ending in the X state (i.e. X1X or 1X1X1....X.).

```
/ PIN NAMES FROM SAMPLE.SR     6-14-79 /
 ASSIGN PNAME  J1PIN5=5,J1PIN6=6,J1PIN7=7,J1PIN8=8;


BEGIN CAPS 'SAMPLE';
SET PS-1 V=5 I=3;
ASSIGN LGC IA(5,7,8,25-28,76)VILA=0.4 VIHA=3.0
           OA(29-50,71)VOLA=1.0 VOHA=1.5;
IGNORE LGC(71);                 /IGNORE PROBE SENSOR/
        IGN LGC (35-50,71);

/1/

10:  IH(26-28,76)
        IL(5,7,8,25)        OH(29-34);              /RESET FF,SET CTR
        CLOCKS & LOAD/
20:  IL(76) IH(5);              /TEST FF SET/
30:  IH(7,76);                  /SET UP CLOCK/
           IGN LGC (35-50,71);
IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);

/10/

IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
   IL(7);
   IH(7,8);
   IL(7);
   IH(7) IL(8);
      ●
      ●

40: IL(26)
        OL(35,38,39,42,43,46,47,50)        OH(36,37,40,41,44,45,48,49);
                                   /LOAD INTO COUNTER/
50: IH(26);
        IGN LGC (71);
60: IL(7);                      /MOVE CHECKERBOARD OVER/
70: IH(7);
80: IL(26)
        OL(36,37,40,41,44,45,48,49)        OH(35,38,39,42,43,46,47,50);
90: IH(26);
100:IH(25)
        OL(35,38,39,42,43,46,47,50);                          /CLEAR COUNTER/
110:IL(25);
120:IL(28)
        OL(30,32,34);
130:IH(28)
        OH(30,32,34-50);

/50/

140:IL(27)
        OL(29,31,33);
150:IH(27)
        OL(35-50)        OH(29,31,33);
FINISH CAPS;
        IGN LGC (71);

END;
```

Figure 5-26. CAPS Digitally Autoprogrammed Test Program (.DA File).

```
 SIMULATION RESULTS FOR SAMPLE.SR AND SAMPLE.TP   6-14-79   PAGE: 1-1

STEPS: 0 - 49
            0           1           2           3           4
            01234 56789 01234 56789 01234 56789 01234 56789 01234 56789
  EXTERNAL INPUTS
J1PIN25   25 X0... ..... ..... ..... ..... ..... ..... ..... ..... .10..
J1PIN26   26 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, 01,,0 1,,,,
J1PIN27   27 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN28   28 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
J1PIN29 B 29 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN30 B 30 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN31 B 31 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN32 B 32 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN33 B 33 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN34 B 34 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN5     5 X01,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN7     7 X0.10 10101 01010 10101 01010 10101 01010 10101 ,,01, ,,,,,
J1PIN76   76 X101, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN8     8 X0... 1,0.1 ,0.1, 0.1,0 .1,0. 1,0.1 ,0.1, 0.1,0 ..... .....

   EXTERNAL OUTPUTS
J1PIN29 B 29 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN30 B 30 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
J1PIN31 B 31 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN32 B 32 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
J1PIN33 B 33 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
J1PIN34 B 34 X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
J1PIN35   35 X---- ----- ----- ----- ----- ----- ----- ----- 0...1 ,0..1
J1PIN36   36 X---- ----- ----- ----- ----- ----- ----- ----- 1,,,0 ....1
J1PIN37   37 X---- ----- ----- ----- ----- ----- ----- ----- 1,,,0 ....1
J1PIN38   38 X---- ----- ----- ----- ----- ----- ----- ----- 0...1 ,0..1
J1PIN39   39 X---- ----- ----- ----- ----- ----- ----- ----- 0...1 ,0..1


   INTERMEDIATES
CDN1            X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
CDN2            X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
CDN3            X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01
CDN4            X1,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,01

   DEVICE PINS
IC1 - 7495
 PIN: IN    1 X0... 1,0.1 ,0.1, 0.1,0 .1,0. 1,0.1 ,0.1, 0.1,0 ..... .....
      IN    2 1,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
      IN    3 1,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
      IN    4 1,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
      IN    5 1,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,, ,,,,,
      IN    6 0.... ..... ..... ..... ..... ..... ..... ..... ..... .....
      IN    8 0.... ..... ..... ..... ..... ..... ..... ..... ..... .....
      IN    9 X0.10 10101 01010 10101 01010 10101 01010 10101 ,,01, ,,,,,
```

Figure 5-27. Good Board Simulation Results (PRETTY).

2. More than one transition between the 0 state and the X state but ending in the X state, (i.e. X0X, or 0X0X0....X.).

3. A continuous oscillation is occuring at this pin at this step (i.e. 10101010...). The oscillation must be greater than 128 micro steps.

   A pin or node that has the R state indicated will, for testing purposes, be considered to be in the X state.

Use of the Single Step expansion feature of the CAPS/APG PREPARE Mode will indicate the exact behavior of a P, N, or R entry.

5.4.4.4 Pin Activation Results (ACTIVE)

A typical pin activation results listing is shown in Figure 5-28. As with the PRETTY listing, the DISPLAY OPTIONS page can be used to generate the type of pin activity list desired (EXTERNAL INPUTS, EXTERNAL OUTPUTS, INTERMEDIATES, and DEVICE PINS can be included in the list or deleted, as required). The listing reflects LOW ACTIVITY and HIGH ACTIVITY for each pin and is compiled directly from the Simulation Results listing (PRETTY file). For example, EXTERNAL INPUT 25 was first driven low at test step 1 (in the PRETTY file) and remained low until test step 46. It went high at 46 but then low again from test steps 47 through 51. It therefore completed three TRANSITIONS being first low at test step 1 and last low at test step 51. It remained low for 50 test steps, and was high at only one test step (46).

As with the PRETTY file, B represents a Bus connection, while the 0, 1, X, P and N states indicate that the pin was at some time during the program in that respective state.

The ACTIVE results are used primarily to show those places in the network where a pin or node NEVER changes state. An indication of NEVER is allowed if followed by an explanation (NC, GND, VCC, etc.). NEVER, followed by a row of asterisks (****), is an indication that the pin never goes either High or Low. It is because the test program does not excercise the pin to that state that fault detection on the pin is not accomplished.

5.4.5 F Sub-Mode Data Outputs

The data outputs listed in the following paragraphs can occur as the result of a successful Sub Mode F run (Fault Detection Simulation). Each output can be suppressed by turning the associated PRINT LISTINGS Option OFF.

5.4.5.1 Fault Class Groups Detected & % Detection (.FG File)

The first data output of a successful Sub mode F run is the Fault Class Groups Detected & % Detection Listing (.FG File). This listing can be used to determine how effective the test program is at exercising a board so that modeled faults are detected (see Figure 5-29). The following data is provided:

1. TEST#. The first test step at which the modeled fault class is detected. Test numbers correspond to CAPS inserted test numbers / 1 / in the .DA File.

2. FAULT CLASS SETS DETECTED. Each fault class set detected at that test step is listed. Fault class sets are identified in the Fault Detection by Class listing (Figure 5-27) by CLASS #. For example, the first fault class detected at TEST #1 is 5. Figure 5-27 lists Fault CLASS 5 as J1 PIN 27 SA0. Fault class sets that provide the same output patterns are grouped together in parenthesis (5,7,9,. .). IF CAPS prints an asterisk followed by a fault class set *(5,7,9,. .) it indicates that one or more of the fault class sets listed could cause indeterminate states and therefore were probably detected at this step. If a *NONE* message occurs, it indicates that no fault classes were detected at the test steps listed.

3. UNDETECTED FAULTS. All fault classes that have been modeled but not detected are listed. After an analysis of this data, additional tests can be added to the test program to exercise the board more completely if the percent detection is not sufficient for the board being simulated.

```
PIN ACTIVATION RESULTS FOR SAMPLE.SR AND SAMPLE.TP   6-14-79
           TEST 0   LOW ACTIVITY            HIGH ACTIVITY
51 TESTS          FIRST  LAST  # TESTS   FIRST  LAST  # TESTS  TRANSITIONS
  EXTERNAL INPUTS
J1PIN25    25 X      1    51     50        46    46       1          3
J1PIN26    26 X     40    44      2         1    51      49          5
J1PIN27    27 X     50    50      1         1    51      50          3
J1PIN28    28 X     48    48      1         1    51      50          3
J1PIN29 B  29 X   NEVER   BUSS              1    51      51          1
J1PIN30 B  30 X   NEVER   BUSS              1    51      51          1
J1PIN31 B  31 X   NEVER   BUSS              1    51      51          1
J1PIN32 B  32 X   NEVER   BUSS              1    51      51          1
J1PIN33 B  33 X   NEVER   BUSS              1    51      51          1
J1PIN34 B  34 X   NEVER   BUSS              1    51      51          1
J1PIN5      5 X      1     1      1         2    51      50          2
J1PIN7      7 X      1    42     21         3    51      30         40
J1PIN76    76 X      2     2      1         1    51      50          3
J1PIN8      8 X      1    51     33         5    38      18         19

   EXTERNAL OUTPUTS
J1PIN29 B  29 X     50    50      1         1    51      50          3
J1PIN30 B  30 X     48    48      1         1    51      50          3
J1PIN31 B  31 X     50    50      1         1    51      50          3
J1PIN32 B  32 X     48    48      1         1    51      50          3
J1PIN33 B  33 X     50    50      1         1    51      50          3
J1PIN34 B  34 X     48    48      1         1    51      50          3
J1PIN35    35 X     40    51      8        44    50       4          5  X
J1PIN36    36 X     44    51      6        40    50       6          4  X
J1PIN37    37 X     44    51      6        40    50       6          4  X
J1PIN38    38 X     40    51      8        44    50       4          5  X
J1PIN39    39 X     40    51      8        44    50       4          5  X
J1PIN40    40 X     44    51      6        40    50       6          4  X

   INTERMEDIATES
CDN1          X     48    48      1         1    51      50          3
CDN2          X     48    48      1         1    51      50          3
CDN3          X     48    48      1         1    51      50          3
CDN4          X     48    48      1         1    51      50          3
CLRCNT        X      1    51     50        46    46       1          3
CLRFF         X      1     1      1         2    51      50          2
CUP1          X     50    50      1         1    51      50          3
CUP2          X     50    50      1         1    51      50          3

   DEVICE PINS
IC1 - 7495
 PIN: IN    1 X      1    51     33         5    38      18         19
      IN    2 1   NEVER   NC                1    51      51          0
      IN    3 1   NEVER   NC                1    51      51          0
      IN    4 1   NEVER   NC                1    51      51          0
      IN    5 1   NEVER   NC                1    51      51          0
      IN    6 0      1    51     51     NEVER   GND                  0
      IN    8 0      1    51     51     NEVER   GND                  0
      IN    9 X      1    42     21         3    51      30         40
      BUSS10 X      10    41     18        12    51      24         16  X
      BUSS11 X       8    51     26        10    41      18         17  X
      BUSS12 X       6    41     20         8    51      26         18  X
      BUSS13 X       4    51     28         6    41      20         19  X
```

Figure 5-28. Pin Activation Results.

4. STATISTICS. This column indicates the total number of fault classes detected at each test divided by the total number of fault class sets that provide different output patterns. For example, TEST #1 has sixteen fault classes that provide five different output patterns (16/5 = 3.20).

5. % DETECT. This column lists the percentage of fault classes that have been detected up to and including the test step listed. The final column figure therefore indicates the total percentage of faults that have been detected by the complete test program. For example, 16.21% faults have been detected up to and including TEST #3. That is, 52 fault classes out of a possible total of 328 fault classes have been detected (52/328 = .1621 = 16.21%). The final figure lists 92.79% faults detected at the last test step (50). That is, 297 out of a possible total of 328 fault classes have been detected (297/328 = .9279 = 92.79%).

```
CAPS FAULT DETECTION FOR SAMPLE.SR AND SAMPLE.TP    6/14/1979   PAGE 1

        FAULTS MODELED FOR THIS RUN:

IC OUTPUTS STUCK AT 0 AND 1

TEST #          FAULT CLASS SETS DETECTED          STATISTICS    % DETECT
-------------------------------------------------------------------------

1       *(5, 7, 9, 11, 13, 15, 84, 86, 96, 98),
        (3), (17, 108), (19, 110), (25)            16/5    3.20     14.41
2       (21)                                       1/1     1.00     15.31
3       (23)                                       1/1     1.00     16.21
4                *NONE*
5       (27)                                       1/1     1.00     17.11
6-39             *NONE*
40      *(4, 24, 29), (2), (28), (30, 52), (31),
        (32, 34), (35, 77), (36), (39, 67), (40),
        (42), (45, 63), (46, 60), (47), (48, 50),
        (51, 89), (54, 68), (55), (56, 58), (59,
        101), (64, 66), (78), (80), (83), (90),
        (92), (95), (102), (104), (107)            44/30   1.46     56.75
41-43            *NONE*
44      *(38, 41, 43, 44, 65, 69), (33, 81), (
        37, 79), (49, 93), (53, 91), (57, 105),
        (61, 103), (62), (76), (82), (88), (94),
        (100), (106)                               25/14   1.78     79.27
45               *NONE*
46      *(1)                                       1/1     1.00     80.18
47               *NONE*
48      (8, 87), (12), (16), (20, 111), (99)       7/5     1.40     86.48
49               *NONE*
50      (6, 85), (10), (14), (18, 109), (97)       7/5     1.40     92.79
51               *NONE*


UNDETECTED FAULTS:

22, 26, 70, 71, 72, 73, 74, 75
```

Figure 5-29. Fault Class Sets Detected and % Detection (.FG File).

5.4.5.2 Fault Classes Modeled (.FP File)

The Fault Classes Modeled (.FP File) is a large listing that is output next (if the PRINT LISTINGS F Option is ON). The format of the file can be changed by the MISCELLANEOUS OPTIONS C, V, I, and U so that either a complete listing or selected sections of the listing can be output, if desired. The following file sections may be output:

1. Fault Detection By Class #. If the MISCELLANEOUS C Option is ON, the listing shown in Figure 5-30 will occur. The listing includes the complete set of simulated faults grouped by CLASS #, test number that the fault class was first DETECTED at (from the .DA File), and EQUIVALENT FAULTS. Before the AFD simulation run, faults are collapsed into groups called Fault Classes. A fault class consists of those modeled faults with output patterns that are inherently indistinguishable at the UUT edge connector pins, due to circuit topology, no matter how many input patterns are applied. Equivalent faults are those faults that are considered equivalent to the single fault being modeled. For example, in Figure 5-30 CLASS #1 and 6 may be read as follows:

   CLASS #1 can be read as "J1PIN25 is SA0. This class was first DETECTED at test step 46 (in the .DA file)".

   CLASS #6 can be read as "J1PIN27 is SA1." The class was first DETECTED at test step 50.

```
 FAULT DETECTION BY CLASS # FOR SAMPLE.SR & SAMPLE.TP 6/14/79 PAGE    1
CLASS# DETECTED EQUIVALENT FAULTS
------ -------- -----------------
   1      46     J1PIN25-0
   2      40     J1PIN25-1
   3       1     J1PIN26-0
   4      40     J1PIN26-1
   5       1     J1PIN27-0
   6      50     J1PIN27-1
   7       1     J1PIN28-0
   8      48     J1PIN28-1
   9       1     J1PIN29-0
  10      50     J1PIN29-1
  11       1     J1PIN30-0
  12      48     J1PIN30-1
  13       1     J1PIN31-0
  14      50     J1PIN31-1
  15       1     J1PIN32-0
  16      48     J1PIN32-1
  17       1     J1PIN33-0
  18      50     J1PIN33-1
  19       1     J1PIN34-0
  20      48     J1PIN34-1
  21       2     J1PIN5-0
  22   ********  J1PIN5-1 IC6.2-0 IC6.4-1
  23       3     J1PIN7-0
  24      40     J1PIN7-1
  25       1     J1PIN76-0
  26   ********  J1PIN76-1
  27       5     J1PIN8-0
  28      40     J1PIN8-1
  29      40     GND.1-1
  30      40     IC1.10-0
  31      40     IC1.10-1
  32      40     IC1.11-0
  33      44     IC1.11-1
  34      40     IC1.12-0
  35      40     IC1.12-1
  36      40     IC1.13-0
  37      44     IC1.13-1
  38      44     IC10.2-0
  39      40     IC10.2-1
  40      40     IC10.3-0
  41      44     IC10.3-1
  42      40     IC10.6-0
  43      44     IC10.6-1
  44      44     IC10.7-0
  45      40     IC10.7-1
  46      40     IC2.10-0
  47      40     IC2.10-1
  48      40     IC2.11-0
  49      44     IC2.11-1
  50      40     IC2.12-0
  51      40     IC2.12-1
  52      40     IC2.13-0
  53      44     IC2.13-1
  54      40     IC3.10-0
  55      40     IC3.10-1
  56      40     IC3.11-0
  57      44     IC3.11-1
```

NOTE

The MISCELLANEOUS C option must be ON for this section of the listing.

Figure 5-30. Fault Detection By Class Number (.FP File).

2. Undetected Faults By Class #. If the MISCELLANEOUS V Option is ON, the listing shown in Figure 5-31 will occur. The listing includes a complete set of those fault classes that were not detected during the simulation run. The listing is similar in content to the detected fault class listings just described. For example, Fault CLASS #22 may be read as "J1 PIN 5 is SA1 and this is equivalent to IC6.2 being SA0 and IC6.4 being SA1."

3. Detected Faults By External & IC Pin. If the MISCELLANEOUS I Option is ON, the listing shown in Figure 5-32 will occur. The listing may be read as follows:

> "EXTernal J1PIN25 is an INput pin." The fault EXT:J1PIN25 SA0 was first detected at test step 46 (in the .DA file) and is assigned to Fault CLASS 1.

```
UNDETECTED FAULTS BY CLASS # FOR SAMPLE.SR & SAMPLE.TP 6/14/79 PAGE    1
CLASS# EQUIVALENT FAULTS
------ -----------------
  22   J1PIN5-1 IC6.2-0 IC6.4-1
  26   J1PIN76-1
  70   IC6.2-1 IC6.4-0
  71   IC6.6-0
  72   IC6.6-1
  73   IC5.5-0
  74   IC5.5-1
  75   IC5.6-0
```

NOTE

The MISCELLANEOUS V option must be ON for this section of the listing.

Figure 5-31. Undetected Faults by Class Number (.FP File).

```
 EXTERNAL FAULT DETECTION FOR SAMPLE.SR & SAMPLE.TP 6/14/79 PAGE    1
EXT     FAULT DETECTED CLASS# EQUIVALENT FAULTS
-----   ----- -------- ------ -----------------
EXT:J1PIN25
IN       SA0      46     1
         SA1      40     2
EXT:J1PIN26
IN       SA0       1     3
         SA1      40     4
EXT:J1PIN27
IN       SA0       1     5
         SA1      50     6
EXT:J1PIN28
IN       SA0       1     7
         SA1      48     8
EXT:J1PIN29
BUSS     SA0       1     9
         SA1      50     10
EXT:J1PIN5
IN       SA0       2     21
         SA1   ******    22     IC6.2-0 IC6.4-1
EXT:J1PIN7
IN       SA0       3     23
         SA1      40     24
```

NOTE

The MISCELLANEOUS I option must be ON for this section of the listing.

```
 IC FAULT DETECTION FOR SAMPLE.SR & SAMPLE.TP 6/14/79 PAGE    1
PIN     FAULT DETECTED  FIRST 0,1  CLASS# EQUIVALENT FAULTS
-----   ----- -------- ----------- ------ -----------------
IC: GND
TYPE: GND
 1  0   SA1     40         1,NEVER  29
IC: IC1
TYPE: 7495
10  0   SA0     40         10,12     30
        SA1     40                   31
11  0   SA0     40          8,10     32
        SA1     44                   33
12  0   SA0     40          6,8      34
        SA1     40                   35
13  0   SA0     40          4,6      36
        SA1     44                   37
IC: IC10
TYPE: 74193
 2  0   SA0     44         40,44     38
        SA1     40                   39
 3  0   SA0     40         44,40     40
        SA1     44                   41
 6  0   SA0     40         44,40     42
        SA1     44                   43
 7  0   SA0     44         40,44     44
        SA1     40                   45
```

Figure 5-32. Detected Faults by Ext. and IC Pin Number (.FP File).

The fault "EXT:J1PIN25 SA1" was first detected at test step 40 and is assigned to Fault CLASS #2.

A series of asterisks (******) in the IC FAULT DETECTION DETECTED column indicate those fault classes that were not detected during the run. The IC Fault Detection Data may be read as:

"IC1 is a Type 7495." IC1.10 is an Output. The fault was first detected at test step 40. The first 0 and 1 states are also listed. For example, IC1.10 first went low at test step 10, and high at test step 12.

4. Undetected Faults By External & IC Pin. If the MISCELLANEOUS U Option is ON, the listing shown in Figure 5-33 will occur. The listing is read in a similar manner to the Detected Faults Listing just described.

5.4.5.3 Selective Fault Data Outputs

If the PRINT LISTINGS SF Option is ON, selected faults associated with the Fault Classes modeled (.FP) file can be run. Prior to running any selective faults, a complete fault simulation run without selective faulting must have been at least started. A Selective Faults (.SF) command file is also required prior to starting a run. The faults that are to be selected and run must be included in the file. Details are contained in Para. 5.2.6.3. Some examples of .FP file selective fault data outputs obtained when the MISCELLANEOUS V and I Options are also ON are shown in Figure 5-34.

The first SF Output reflects all undetected faults associated with IC1, IC2, and IC10 only. The second SF Output reflects all undetected IC Faults for IC1, IC2, and IC10 only. Any combination of selected faults may be output using the procedures contained in Para. 5.2.6.3 if the associated PRINT LISTINGS and MISCELLANEOUS Options are also ON.

5.5 PERFORMING A TYPICAL RUN

Once the network description and basic test program have been written using the procedures in Section 2, the following steps can be taken to create, evaluate, improve, and debug both files:

1. Use EDIT to create the network description .SR file and the basic test program .TP file (refer to Para. 5.5.2).

2. Use the N sub-mode of PREPARE to check the network description .SR file for syntax and other errors. Warnings, statistics, and cross-reference data may also be listed at this time.

3. If the APG Monitor is being used, the I sub-mode of PREPARE can be entered at this time and the test program prepared "on-line" in increments. Each test program increment can then be checked by incremental simulation and either accepted (appended to the test program) or rejected (deleted) depending on the incremental simulation results. Details are contained in Appendix B.

4. If the APG Monitor is being used, it will also be possible to use the equation generation and track connectivity sections of the E/T/C Mode at this time to help "debug" the test program if required (refer to Appendix B).

```
 UNDETECTED EXTERNAL FAULTS FOR SAMPLE.SR & SAMPLE.TP 6/14/79 PAGE   1
EXT    FAULT CLASS# EQUIVALENT FAULTS
----- ----- ------ -----------------
EXT:J1PIN5
IN      SA1   22     IC6.2-0 IC6.4-1
EXT:J1PIN76
IN      SA1   26
```

```
 UNDETECTED IC FAULTS FOR SAMPLE.SR & SAMPLE.TP 6/14/79 PAGE   1
PIN    FAULT  FIRST 0,1  CLASS# EQUIVALENT FAULTS
----- ----- ----------- ------ -----------------
IC: IC5
TYPE: 7474
 5  O   SAO      1,2      73
        SA1               74
 6  O   SAO      NC       75
IC: IC6
TYPE: 7404
 2  O   SAO      2,1     (22)
        SA1               70     IC6.4-0
 4  O   SAO      1,2     (70)
        SA1              (22)
 6  O   SAO      3,1      71
        SA1               72
```

NOTE

The MISCELLANEOUS U option must be
ON for this section of the listing.

Figure 5-33. Undetected Faults by Ext. and IC Pin Number (.FP File).

```
        ********  SELECTIVE FAULT SIMULATION RESULTS   ********
  UNDETECTED FAULTS BY CLASS # FOR SAMPLE.SR & SAMPLE.TP 3/9/79 PAGE 1
CLASS#  EQUIVALENT FAULTS
260     IC1.1 SHORTED TO IC1.2
 65     IC1.2-0
 66     IC1.3-0
 67     IC1.4-0
 68     IC1.5-0
 70     IC1.8-1
274     IC10.13 SHORTED TO IC10.14
108     IC2.3-0
```

```
        ********  SELECTIVE FAULT SIMULATION RESULTS   ********
NO UNDETECTED EXTERNAL FAULTS FOR SAMPLE SR & SAMPLE TP 3/9/79
```

```
        ********  SELECTIVE FAULT SIMULATION RESULTS   ********
  UNDETECTED IC FAULTS FOR SAMPLE. SR & SAMPLE. TP 3/9/79 PAGE 1
PIN     FAULT   FIRST 0.1   CLASS#  EQUIVALENT FAULTS
IC:   IC1
TYPE:   7495
  1 I   S- 2      1.4       260
  2 I   SAO       NC        65
  3 I   SAO       NC        66
  4 I   SAO       NC        67
  5 I   SAO       NC        68
  8 I   SA1       1,NEVER   70
IC:   IC10
TYPE:   74193
  130   S-14      NC        274
IC:   IC2
TYPE:   7495
  3 I   SAO       NC        108
```

Figure 5-34. Selective Fault Data Outputs.

5. The T sub-mode of PREPARE can be used with either monitor (APG or CAPS) to check the basic test program .TP file for syntax and other errors, expand test program macro statements, re-number labels, and generate a test program data base required for simulation.

6. The G sub-mode of PREPARE can be used with either monitor (APG or CAPS) to simulate a good board, automatically insert output patterns into the test program (.DA file) and list any differences between outputs included in the .TP file and simulated outputs inserted into the .DA file. Once the run has been completed, the Utility Program DISPLAY can be used to output all or selected segments of the Good Board Simulation Results. This listing can be used to determine if the simulated circuit is being initialized properly, if logic is being exercised, whether pulses or races have occurred, or whether unknown states exist. A test technician can rapidly determine if the basic test program is adequate or needs to be expanded with such an analysis.

7. If the APG Monitor is being used, the Activity Tracking section of E/T/C Mode can be used at this time to help track down such problems as "X" states, positive or negative going pulses, and level changes occurring on active pins listed in the Good Board Simulation Results (PRETTY). Details are contained in Appendix B.

8. The D sub-mode of PREPARE can be used with either monitor (CAPS or APG) to create the guided probe data base (.PD file) and the binary data bases required for device testing in the DIAGNOSE Mode (.TA file for 1792 A/B systems, .TD file for 1792 C/D systems, .TV file in 1796 test systems, or .TL file for 1795, 1799 systems). The P sub-mode is also entered automatically at this time so that the user furnished probe starting point (.US), jumping point (.UJ) files, User Discrimination (.UD) files and Conditional-Connectivity files can be translated, if available. After D has been run, the MANUAL TESTING, and GUIDED PROBE/FINDS options of the DIAGNOSE Mode can be used. The AUTOMATIC FAULT LOCATION option cannot be run until the F sub-mode of PREPARE has been run. If the Quick Dictionary Lookup feature is to be used, then the Q sub-mode of PREPARE must be run after running the F sub-mode (refer to para 5.2.7).

9. Next, the DIAGNOSE Mode should be entered and a known good board used to determine if any mis-matches exist between the CAPS/APG simulation data and physical circuit. A PASS indication signifies that the simulation run has been successful. A FAIL indication signifies a mis-match between the physical board under test and the simulation data generated by CAPS or APG. Some possible reasons for the mis-match might include:

1. Schematic diagram errors.

2. Errors in the network description (.SR file).

3. An actual fault has been detected on the known good board (the board is not fault-free as it should be).

4. Use of illegal states or inputs by the test program.

5. Wiring errors in the test system device adaptor used.

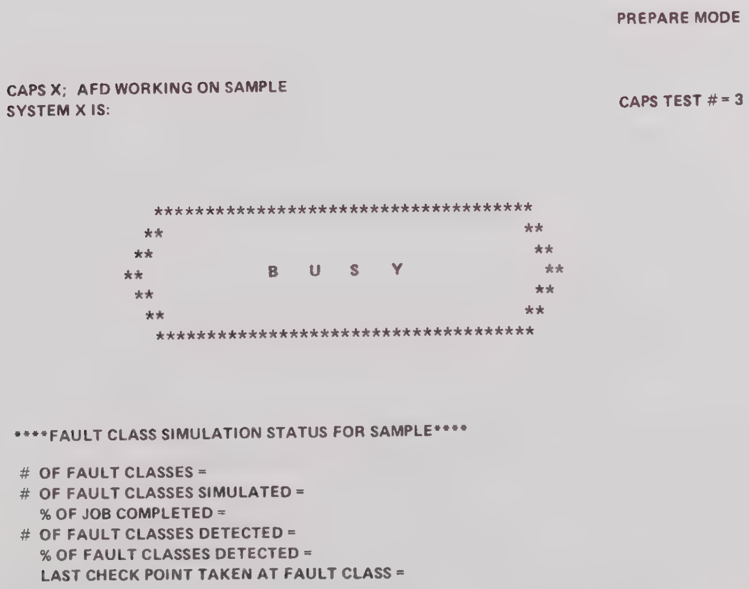
6. Incorrect IC models have been inserted into the CAPS or APG User Library.

The computer guided probe may be used at this point to rapidly determine the point at which the simulation data and physical board first disagree.

10. Next, the F-sub-mode of PREPARE can be used to model faults and generate data to evaluate the fault detecting capability of the test program. During the initial run, it is usually sufficient to model stuck faults at output pins only (the Fault Modeling 0 Option is ON). This is a fast method of fault simulation because it does not require that Input faults be simulated. Only those failures that correspond to an entire signal line being SA0 or SA1 are simulated, thus providing a rapid indication of how effective the test program is in exercising a board (clocking data through a board, enabling circuits, etc.). Each IC Output pin of the good model must change state in order for this type of failure to be detected.

11. During the AFD run, the status message shown in Figure 5-35 will occur. The message is updated automatically as the run progresses (after every six fault classes have been simulated) thus reflecting the current status of the run. The next to last message entry (LAST CHECK POINT TAKEN AT FAULT CLASS=) reflects the last group of six fault classes simulated. If the Teletype CTRL X keys are activated during a run, the AFD simulation will be terminated and the CAPS or APG Monitor entered (regardless of whether the RESTART option is ON or OFF.) By turning the RE-START option ON, the fault simulation can be re-started at or near the point of termination.

12. Next, data outputs from the fault modeling run are evaluated to determine how good the test program is in detecting faults. The Utility Program LISTALL can be used to output all or selected segments of the .FG and .FP file listings after completing a run. The .FG file lists the percentage of faults detected by the test program while the .FP file lists



```
                                              PREPARE MODE


CAPS X;  AFD WORKING ON SAMPLE
SYSTEM X IS:                                  CAPS TEST # = 3



          **************************************
         **                                    **
        **                                      **
       **            B   U   S   Y               **
        **                                      **
         **                                    **
          **************************************



****FAULT CLASS SIMULATION STATUS FOR SAMPLE****

 # OF FAULT CLASSES =
 # OF FAULT CLASSES SIMULATED =
   % OF JOB COMPLETED =
 # OF FAULT CLASSES DETECTED =
   % OF FAULT CLASSES DETECTED =
   LAST CHECK POINT TAKEN AT FAULT CLASS =
```

Figure 5-35. Prepare Mode AFD Status Message.

undetected faults by class number. The basic test program (.TP file) may have to be improved at this point to increase the percentage of detection to acceptable limits thus necessitating another T and G mode run. Always improve the basic test program (.TP file) when required, not the digitally autoprogrammed test program (.DA file). The selective faulting feature can be turned on at this point to re-run the test program against user selected faults only. This feature makes it possible to determine if the modified test program will now detect those failures that were missed by the original test program, and it will do this in a small fraction of the time required for a complete fault simulation run.

13. Once the test program has been improved sufficiently to detect an acceptable percentage of stuck faults at output pins (usually after two or three runs), all faults may be modeled. Prior to entering the DIAGNOSE Mode for device testing, it is suggested that a final run be performed with all sub-mode options that are needed turned on, and without selective faulting. The final run will insure that all inconsistencies have been eliminated and that the correct data bases for device testing in the DIAGNOSE Mode have been generated.

14. If the Quick Dictionary Lookup feature is to be used in the DIAGNOSE Mode (instead of AFL) to pass starting points to the Probe, then the Q sub-mode of PREPARE should be run at this time in order to generate the necessary data base files (refer to Para. 5.2.7).

Each of the steps listed can be performed by entering each mode and setting the options for that mode to the desired states. Next, a monitor SAVE command can be issued to save the states of the mode. The states saved in this manner may be easily invoked through command strings issued from the CAPS or APG Monitor at any time (refer to Para. 3.3.6). The advantage of this approach is that the user can establish the options that he or she desires for each run.

5.6 FILE MANAGEMENT

5.6.1 General

A pre-recorded system cartridge is supplied with each CAPS or APG equipped test system for system use. The cartridge has CAPS or APG system files, system monitor files, and may be used by CAPS or APG to store some test and listing files on the SYS: side. The opposite side (DSK1:) is used to store CAPS or APG test and listing files together with a rather extensive CAPS IC Library. The CAPS User Library is also stored on the DSK: side. The User Library (DSK:USERLB.CL) contains user inserted devices while the IC Library (DSK:CAPSX.CL) includes a series of devices extracted from various IC manufacturer's catalogs.

CAPS or APG Listing files are ASCII text files that are created by CAPS or APG during PREPARE and GENERATE Mode runs. Both listing and test files are identified in Table 5-11 and are automatically assigned by CAPS or APG to a specific working disk as they are created. If double disks are available, or CAPS/APG files may also be created on the additional disk devices through use of the WORKING DISKS (W=), or EXTENDED WORKING DISKS (X=) Options. Details are contained in Para. 5.3.2.6. Two other ASCII files, the network description (.SR file) and basic test program (.TP file), are created by the user through use of EDIT and may be stored on any disk that has sufficient room.

CAPS or APG test files are binary files that are created automatically and contain test information and other data that is required for CAPS or APG operation.

During a CAPS or APG run, a series of temporary intermediate files are also created. Intermediate files contain temporary data that is required only for the module or program that is running. During clean-up operations at the end of a run, the files can be deleted automatically. If a CAPS or APG run is terminated before the clean-up phase, however, (by executing a CTRL C for example) some of these files may appear in device directories.

The following paragraphs describe how to:

1. Prepare a second system cartridge for use, and format the cartridge using the Utility Programs ZRODSK or RKFORM when required (refer to Para. 5.6.2).

2. Automatically save CAPS or APG test and listing files on a second cartridge using the FILE USAGE Page W= Options so that additional room will be available for simulation runs (refer to Para. 5.6.3).

3. Create a cartridge that can be used for device testing in the DIAGNOSE Mode, but cannot be used to run the PREPARE and GENERATE Modes. The cartridge may be returned to full operational status at any time (refer to Para. 5.6.4).

4. Output all or selected CAPS or APG listing files after completing a PREPARE or GENERATE Mode run using the Utility Program LISTALL (refer to Para. 5.6.5).

5. Look at any CAPS or APG listing file or other ASCII source file from within the CAPS or APG Monitor using the Utility Program LOOK (refer to Para. 5.6.6).

6. Output a complete set of Good Board Simulation Values (PRETTY) or segments of PRETTY printouts only. The sections desired, range of tests desired, and IC and/or external pins desired can be specified. DISPLAY may be run after completing a successful G or I Mode run (refer to Para. 5.6.7).

7. Delete those CAPS or APG test, listing, and temporary files that are no longer required to be saved as files using the Utility Program CLEANUP (refer to Para. 5.6.8).

5.6.2 Preparing a Second Cartridge For Use

A second cartridge can serve the following purposes:

1. It can be used as a "back-up" system cartridge. If the original cartridge is destroyed, lost, or damaged, a "back-up" will be immediately available.

2. It can provide additional room for a CAPS or APG simulation run in the PREPARE Mode by saving selected test and listing files on the bottom cartridge.

3. It can serve as a diagnostic only cartridge. Diagnostic cartridges are used for device testing in the DIAGNOSE Mode. The PREPARE and GENERATE Modes cannot be entered.

To prepare a second system cartridge, proceed as follows:

a. Start-up the test system and install the CAPS/APG pre-recorded system cartridge in the top drive and the second cartridge in the bottom drive.

b. If a fresh (non file structured) cartridge is being inserted for the first time, use the Utility Program RKFORM (for DEC or Diablo drives with DEC or Plessey controllers) or ZRODSK (for Diablo drives and DSD controllers) to structure the cartridge before proceeding. GR supplied cartridges have been file structured and do not require this operation. RKFORM procedures are described in Para. 5.6.2.1 while ZRODSK procedures appear in Para. 5.6.2.2.

c. Enter PIP and type the following command lines to Zero the device directories of the second cartridge and transfer the system header (boot loader, software device handlers, and miscellaneous overhead files) from the top cartridge to the bottom cartridge:

```
.R PIP ↲
*DSK2:<SYS:/Z/Y ↲
*DSK3:</Z ↲
```

d. Enter the keyboard monitor (CTRL C) and use the CCL SQUISH command to transfer all files from the top cartridge to the bottom cartridge in the following manner:

```
.SQ DSK2:<SYS: ↲
.SQ DSK3:<DSK1: ↲
```

SQUISH invokes the GR VPIP /V/S options to complete the transfer. VPIP will transfer files longer than 256 blocks (such as DSK:CAPS8.CL) while PIP will not. The keyboard monitor is entered automatically when the transfer has been completed.

5.6.2.1 RKFORM (Formatting DEC or Diablo Drives With DEC or Plessey Controllers)

The Utility Program RKFORM can be used to zero (clear) a cartridge and then format it into a file structured device of 256 word blocks so that read or write operations can be accomplished from any desired block. To use RKFORM, proceed as follows:

a. Make sure that the cartridge receiving data is installed in the bottom disk drive. The top drive must contain the CAPS or APG system cartridge.

b. Enter the keyboard monitor and type the following command line to call RKFORM into core for use:

```
.R RKFORM ↲   or    .R FORM1 ↲   (If formatting disk in bottom drive)
```

c. The following printout should then occur:

```
RK8E DISK FORMATTER PROGRAM
FOR ALL QUESTIONS, ANSWER Y FOR YES OR N FOR NO
```

d. Next, the operator is asked to specify the disk drives that are to be formatted:

Printout	Answer	Comments
FORMAT DISK 0?	N	No *must* be specified
FORMAT DISK 1?	Y	Yes *must* be specified
FORMAT DISK 2?	N	
FORMAT DISK 3?	N	
ARE YOU SURE?	Y	Yes must be specified to format disks.

An operator must answer Y (Yes) or N (No) to each question as it is asked. Note that only DISK 1 can be formatted *if* the CAPS or APG system cartridge is installed in the *top* drive (DISK 0). RKFORM will format cartridges installed in *any* drive by answering Y (Yes) to the proper question as it is asked. Disk assignments are shown in Figure 5-36.

CAUTION

> If DISK 0 is formatted, and the CAPS or APG system cartridge is installed, it *will* be zeroed.

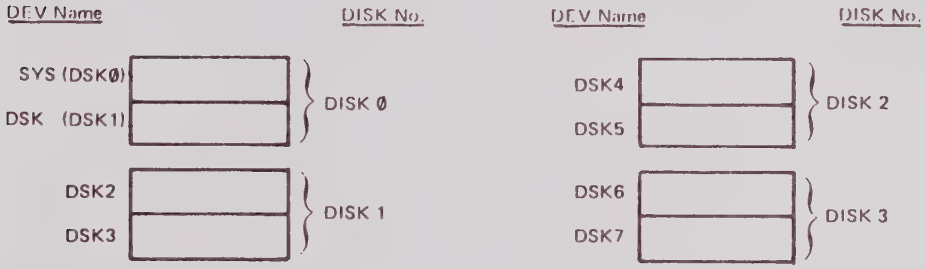


Figure 5-36. RK8E Disk Number Assignments.

e. When a Yes is typed to the final ARE YOU SURE? question, the disk drives specified will be formatted. Once formatting is complete, the RK8E DISK FORMATTER PROGRAM printout described in step c again occurs. To exit the program, press and raise the computer HALT switch. Next, perform a system start-up and enter the keyboard monitor.

5.6.2.2 ZRODSK (Formatting Diablo Disks With DSD Controllers)

The Utility Program ZRODSK can be used in test systems equipped with the Diablo double disk option and a DSD Controller to zero (clear) any cartridge and then format it into a file structured device of 256 word blocks so that read or write operations can be accomplished from any desired block. To use ZRODSK, proceed as follows:

a. Make sure that the cartridge receiving data is installed in the *bottom* disk drive. The *top* drive *must* contain the CAPS or APG system cartridge.

b. Enter the keyboard monitor and type the following command line to call ZRODSK into core for use:

```
.R ZRODSK ↵
```

The following warning message will then occur:

```
ZRODSK WIPES ALL FILES ON DSK2 + DSK3.  ARE YOU SURE?
```

To proceed, depress the Y (Yes) key. Depressing any other key causes an immediate return to the keyboard monitor. If Y is depressed, ZRODSK will automatically zero and format the bottom cartridge and then enter the keyboard monitor.

5.6.3 Saving CAPS or APG Files on a Second Cartridge

CAPS or APG test and listing files that are saved on the pre-recorded system cartridge can be generated on a second cartridge (in double disk systems). This allows more space for the resulting files. CAPS or APG test and listing files are automatically generated on the devices specified by the W= option on the PREPARE Mode FILE USAGE Page (refer to Para. 5.3.2.6). To assign the test and listing files to a second cartridge, enter the PREPARE Mode, and change the FILE USAGE page W= option to read:

```
W=DSK2:,DSK3:
```

When the PREPARE Mode is run, the files will automatically be saved on the devices specified. Other modes and Utility Programs such as DIAGNOSE, GENERATE, TRANSLATE, CLEANUP, and LISTALL must also have the W= option changed to reflect the correct WORKING DISK assignments prior to running the mode or program.

5.6.4 Creating a Testing Only Cartridge

Two methods of preparing a testing only cartridge are available. A user may elect to:

1. Use the second system cartridge as a testing only device. Using this approach, a user creates a testing only cartridge that can easily be returned to full operational status at any time as all system files are present. Once a successful PREPARE Mode run has been completed with the CAPS or APG files saved on the second cartridge, the DEC ODT program may be used to alter the monitor save (.SV) files so that only the DIAGNOSE, SETUP, BATCH, OS, HELP, and LOOK modes can be used. Since PREPARE, GENERATE, CLEANUP, etc. cannot be entered, the CAPS or APG test files cannot be inadvertently destroyed through operator intervention. The full power of the CAPS or APG Monitor can be returned at any time through the use of the DEC ODT program. Special files are also available on the DSK1: device that will automatically make and unmake testing only cartridges using the UPDATE Program. Details are provided in Para. 5.6.4.1.

2. Create a dedicated cartridge that can be used only to run the DIAGNOSE, SETUP, BATCH, OS, HELP, and LOOK modes, but cannot be easily converted back to full operational status. The cartridge will contain only those system save (.SV) files that are needed for device testing and some limited file management, thus providing a maximum amount of room for testing files. Details are provided in Para. 5.6.4.2.

5.6.4.1 Using ODT (or UPDATE) to Change a Complete System Cartridge

Once a successful PREPARE Mode run has been completed and the necessary test files saved, either the DEC ODT program can be used to alter monitor modules so that only the DIAGNOSE, SETUP, BATCH, OS, HELP, and LOOK Modes can be run or an UPDATE run can be made.

The procedure for using ODT is as follows:

a. Enter the keyboard monitor and use the GET command to load the system module to be modified into memory. For example,

```
.GE SYS: MONITR.SV ↙
```

would load the CAPS MONITR.SV module into memory for modification. Each of six modules must be called into core individually using GET, modified with ODT, and then saved using the SAVE command. The following modules must be changed in this manner:

CAPS/APG MONITOR
MONITR.SV
MON1.SV
MON2.SV
MON3.SV
MON4.SV
MON5.SV

b. After loading the module to be changed into memory, use ODT to modify the same location in each module as follows:

```
.GE SYS: NAME.SV ↙

.OD ↙

14043/0004   ↙
```

This calls ODT and changes the content of location 4037 in Field 1 from 0020 to zero. The slash character opens the location and the contents of that location (0020) are listed followed by a space. The new contents are then specified (zero) and the location is closed (↙).

c. Next, exit the ODT program (CTRL C) and use the keyboard monitor SAVE command to save the altered module on the system device. For example,

```
.SA SYS: MONITR.SV ↙
```

will save the modified module on the system device. Repeat steps a through c until all of the CAPS or APG modules listed have been modified and saved.

d. When it is desired to again use all of the CAPS/APG modes (PREPARE, GENERATE, etc.) each module may be called back into core and ODT again used to change the content of location 4037 in Field 1 back to 0020 as in the following example:

```
.GE SYS: MONITR.SV ↙
.OD ↙

14043/0004      ↙
```

When all of the CAPS or APG Modules have been changed, the full power of the monitor will again be available for use.

The file UPDATE.HL is available to help the user run UPDATE. You must have 128 free blocks on DSK: as a "scratch area" before UPDATE can be run.

The procedure for using the UPDATE Program is as follows:

UPDATE is a GR program that can be used to automatically insert software patches into monitor save (.SV) files. The following UPDATE Input source files are available on DSK1: for use with this program:

File Name	Function
CTONLY.ON	Makes a CAPS Testing Only Cartridge.
CTONLY.OF	Turns all CAPS Modes ON again.
ATONLY.ON	Makes an APG Testing Only Cartridge.
ATONLY.OF	Turns all APG Modes ON again.

The UPDATE Input files listed assume that the save (.SV) files to be updated are located on a user-defined device named OUT. Prior to running UPDATE, the OS/8 DEASSIGN and ASSIGN commands must be used to first remove any user-defined names that the system may be using, and second assign the device OUT as the user-defined name to be used. The following OS/8 commands will accomplish this task:

```
.DE ↓
.AS DSK0:OUT ↓
```

The UPDATE Input files can now be run with UPDATE as the following examples indicate:

```
.R UPDATE ↓
*DSK1:CTONLY.ON ↓
   or
*DSK1:CTONLY.OF ↓
```

The input file is first translated, and if no errors are encountered, the patches are automatically inserted into (or removed from) the save (.SV) files listed in step a.

When finished with UPDATE, the user-defined name must be deleted through use of the DEASSIGN command.

```
.DE ↓
```

Error messages generated by UPDATE, together with a description of the program are contained in the SYS: file UPDATE.HL. The HELP file may be output using the CCL LIST command as the following example indicates:

```
.LI LPT:<DSK:UPDATE.HL ↓
```

5.6.4.2 Creating a Dedicated Testing Cartridge

A dedicated cartridge can be created that will contain only those ystem save (.SV) files that are needed for device testing and some limited ile management. The cartridge provides a maximum amount of room for storage f testing files, but is not easily converted back into a full operational ystem cartridge. To create the cartridge, proceed as follows:

a. Insert a CAPS or APG system cartridge into the top disk drive, start-up the test system, and enter the keyboard monitor (refer to Section 4).

b. Place a formatted cartridge in the bottom drive, and enter the RUN mode (start-up the drive).

c. One of the following OS/8 Batch Jobs may be selected:

APGMTO.BI	For APG Monitor Users
CAPMTO.BI	For CAPS Monitor Users

Select and run either the APG or CAPS OS/8 Batch Job by entering one of the following CCL commands:

```
.SUB SYS: APGMTO/T/E ↓
     or
.SUB SYS: CAPMTO/T/E ↓
```

During either the APGMTO or CAPMTO Batch jobs (due to the presence of MSG commands), the operator must depress the space bar to acknowledge the message commands for the program to continue.

Each batch job will zero DSK2: and DSK3:, copy over a new system head, copy over some OS/8 utility programs, and finally copy over the CAPS/APG system save (.SV) files that will be needed for testing. Next, either CTONLY.ON or ATONLY.ON will be run to automatically patch the system save (.SV) files.

d. To proceed, enter the command

```
.R QUIT↓
```

When an up-arrow (↑) occurs, set both disk drives to LOAD.

e. When the LOAD lamps light, move the cartridge in the lower drive to the top drive, and start-up the drive. When the RUN lamp lights, depress any Teletype key (space bar, etc.) to enter the keyboard monitor. DATE the system again at this time (using the .DA command).

f. Place a cartridge containing all of the board files generated by a CAPS/APG PREPARE Mode run in the lower drive, and start-up the drive.

g. When the RUN lamp lights, enter the command:

```
.MUNG DSK1:TSTDSK,SRNAME ↓
```

Where SRNAME is the name of the board to be tested. This procedure will create the OS/8 Batch Job TMOVE.BI on SYS: which copies all of the CAPS/APG testing only files from DSK2: to DSK0: and from DSK3: to DSK1:.

h. Next, run the OS/8 Batch Job TMOVE.BI as follows:

```
.SUB SYS:TMOVE/T/E ↓
```

During the TMOVE Batch job entry (due to the presence of $MSG commands), the operator must depress the space bar to acknowledge the message commands for the program to continue.

The Teletype printout should be inspected as the batch job is run to see if any transfers did not take place. If any ".VERI" line is followed by a "CANT FIND" message, then none of the files in that ".VERI" command line were transferred. If any of the first four ".VERI" commands were unsuccessful, then the lower cartridge has an incomplete PREPARE Mode run and should be checked. The last six ".VERI" command lines transfer one file each. Some of these files are not necessary in the DIAGNOSE Mode unless a particular option is called for (refer to Table 5-12).

i. If the Diagnosis Logging Option is to be used, then the .DL file must be copied to the device which is specified to be used during Diagnosis. This can be done by entering the command:

```
.VERI DSK0:< DSK2:NAME.DL/N/O ↓
```

Where DSK0: is the device where the .DL file is expected to be found, and DSK2: is where the .DL file was created. NAME represents the board name. In a similar manner the User Subroutine File (UR=NAME) must be copied to the appropriate place. Now the cartridge in the upper drive is ready for board testing.

j. If a board has multiple CAPS sections, then steps f through i should be repeated for each section. Similarly, if more than one board is to be tested from the same cartridge, steps f through i must be repeated for each board. Note also that steps f through i can be used to transfer or back-up the testing files of any board.

Table 5-12

CAPS/APG FILES NEEDED FOR A TESTING DISK

Test Files Needed (DSK2: or DSK3:)

.TA	Binary Test Data For 1792A/B Systems
.TD	Binary Test Data For 1792C/D Systems
.TL	Binary Test Data For 1795/1799 Systems
.TV	Binary Test Data For 1796 Systems
.UR	User Subroutine File (if used)

First Working Device (DSK2:)

.CH	Fault Chain Table(1)
.CM	Compare File(1)
.CT	Correspondence Table(1)
.FU	User Fault Table(1)
.GV	Good Value File
.QL	Quick Lookup Fault Dictionary File (if used)
.SH	Short .ST
.I5	Pulse Discrimination File (if used)
.ST	Simulation Table Data Base (2)

Second Working Device (DSK3:)

.FC	Fault Chains File(1)
.FO	Faulty Outputs File(1)
.I1	Starting Points (if used)
.I2	Jumping Points (if used)
.I3	Probe Interaction
.I7	Conditional-Connectivity Data Base (if used)
.LT	Label Table
.PD	Probe Data Base
.I6	Probe Messages File
.DL	Diagnosis Logging File (on Diagnosis Logging Device - if used)

(1) Test files needed for AFL only.

(2) The .ST file is not needed on the testing disk, however, it should be kept with the other source files It is required if Z mode or P mode are run.

5.6.5 LISTALL

The Utility Program LISTALL may be used to output all or selected data outputs that are available after having completed a successful PREPARE Mode run. The program may be entered from the CAPS or APG Monitor using the various approaches described in Section 3. For example,

# LIS and XMIT

will cause the first page of LISTALL to be displayed (see Figure 5-37). The first page is a FILE USAGE page that must be set to reflect the following information:

1. The name and location of the NETWORK DESCRIPTION FILE that has been run through the PREPARE Mode. For example,

   N=DSK3:SAMPLE.SR

   specifies the file SAMPLE.SR on DSK3:. If a device and file name extension are not specified, DSK: and the .SR extension are assumed. It is therefore only necessary to specify

   N=SAMPLE

   if the file SAMPLE.SR has been saved on DSK:.

2. The name and location of the TEST PROGRAM file used. For example,

   T=DSK3:SAMPLE.TP

   specifies the file SAMPLE.TP on DSK3:. If a device and file name extension are not specified, DSK: and the .TP extension are assumed. It is therefore only necessary to specify

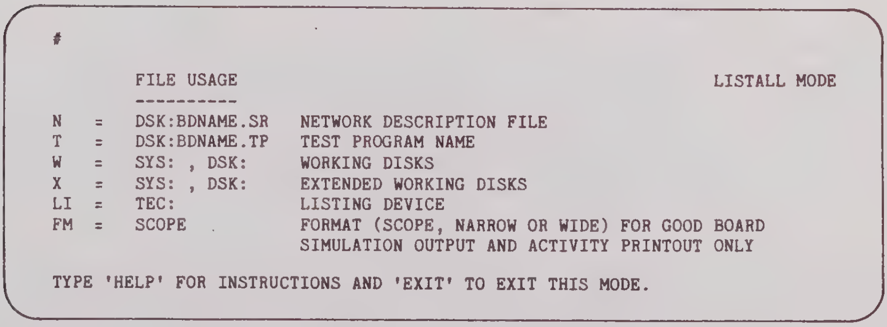
   T=SAMPLE

   if the file SAMPLE.TP has been saved on DSK:.

3. The WORKING DISKS used. During a PREPARE Mode run, CAPS or APG automatically assign test and listing files to the devices specified by W= and X=, respectively.

NOTE

> Once W= and X= have been specified in any mode, the change is automatically made, and applies to all applicable modes.



```
#

        FILE USAGE                                          LISTALL MODE
        ----------
N   =   DSK:BDNAME.SR   NETWORK DESCRIPTION FILE
T   =   DSK:BDNAME.TP   TEST PROGRAM NAME
W   =   SYS: , DSK:     WORKING DISKS
X   =   SYS: , DSK:     EXTENDED WORKING DISKS
LI  =   TEC:            LISTING DEVICE
FM  =   SCOPE           FORMAT (SCOPE, NARROW OR WIDE) FOR GOOD BOARD
                        SIMULATION OUTPUT AND ACTIVITY PRINTOUT ONLY

TYPE 'HELP' FOR INSTRUCTIONS AND 'EXIT' TO EXIT THIS MODE.
```

Figure 5-37. The LISTALL Mode FILE USAGE Page.

4. The LISTING DEVICE to be used. For example,

LI=LPT:

specifies the line printer. The following devices may be listed:

LI=LPT:	Outputs listings to the line printer.
LI=TEC:	Outputs listings to the TEC Display.
LI=TTY:	Outputs listings to the Teletype.
LI=DSK:	Outputs certain listings to DSK1: and saves them in ASCII files named ACTIVE.LS and PRETTY.LS. The files may be output later through use of the LOOK Mode or any of the other file management CCL Commands or Programs (COPY, VERI, PIP, VPIP, etc.).

5. The listing FORMAT to be used. This option applies only when the Good Board Simulation Values (PRETTY), and Pin Activity (ACTIVE) listings are to be output. It has no effect on the other PREPARE Mode listings which are to be output with the possible exception of the .XR file. If the .XR file has to be created now, FORMAT will control the format of this file. Other listings are output in the format that was used at the time they were created during the PREPARE Mode run. The total character width and length that can be used for the PRETTY and ACTIVE listings may be varied as follows:

Command	Maximum Width	Maximum Length
SCOPE	72 characters	24 lines
NARROW	72 characters	60 lines
WIDE	128 characters	60 lines

If TEC display outputs are desired, always specify SCOPE.

FM=SCOPE

If Teletype data outputs (or line printer with 8½ x 11" paper) are desired, use NARROW.

FM=NARROW

If wide paper line printer data outputs are desired, either NARROW or WIDE will work, however, WIDE is recommended for the most efficient utilization of printer paper.

FM=WIDE

Once the FILE USAGE page options have been established, depressing the XMIT key will cause the second LISTALL page (PRINT LISTINGS) to be displayed. The PRINT LISTINGS page is identical to the PREPARE Mode PRINT LISTINGS page shown in Figure 5-16. All or selected listings may be output by turning ON the appropriate options listed on this page (even if the options selected were OFF during the actual PREPARE Mode run). The listings that may be obtained are described further in Para. 5.4.

Depressing the XMIT key again will cause the third and last LISTALL page (DISPLAY OPTIONS) to be displayed. The DISPLAY OPTIONS page is similar to the PREPARE Mode DISPLAY OPTIONS page shown in Figure 5-20 with the exception that the SINGLE STEP EXPANSION options are not included as SINGLE STEP EXPANSION listings cannot be output through LISTALL. Options on the page refer to the Good Board Simulation Results (PRETTY) and PIN ACTIVITY (ACTIVE) listings only, and are described in Para. 5.3.8.

Once all options have been set to output the desired set of listings, the RUN command can be issued to run the program.

# RUN and XMIT

The CAPS or APG Monitor is again entered after the listings specified have been output.

5.6.6 LOOK

The Utility Program LOOK may be used to look at any ASCII source file from within any monitor (CAPS, APG or TEST). The program can be entered from the monitor using any of the various approaches described in Section 3. For example,

# LOO and XMIT

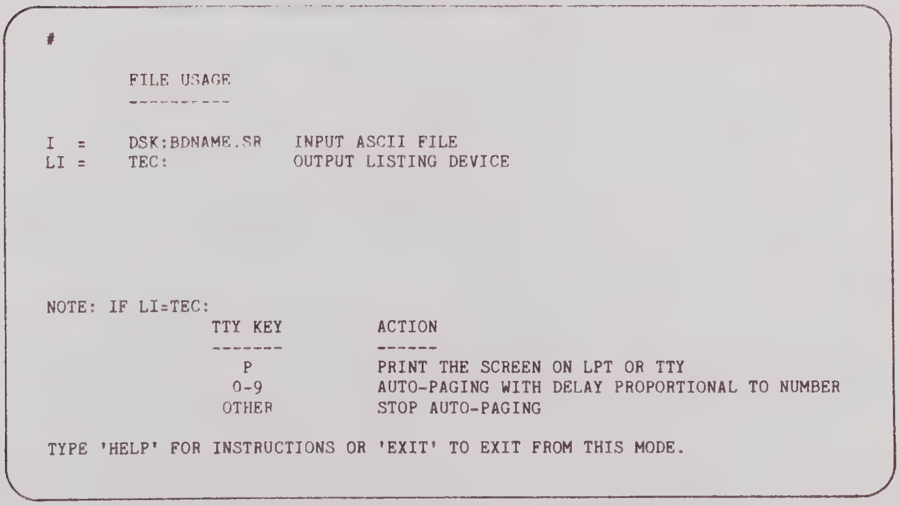
will cause the first and only LOOK page to be displayed (see Figure 5-38). The two options on the page allow the desired ASCII INPUT FILE (I=), and OUTPUT LISTING DEVICE (LI=) to be specified. Any ASCII file located on any file structured device may be listed. If a device is not specified, DSK: is assumed.

Files may be output by changing the I= option to list the file name and device. For example,

I=DSK2:SAMPLE.TP

specifies that the test program SAMPLE.TP located on DSK2: is to be output to the listing device named. Next, the OUTPUT LISTING DEVICE is specified. The following devices may be listed:

LI=LPT:	Outputs listings to the line printer.
LI=TEC:	Outputs listings to the TEC display.
LI=TTY:	Outputs listings to the Teletype.



```
#

        FILE USAGE
        ----------

I  =    DSK:BDNAME.SR    INPUT ASCII FILE
LI =    TEC:             OUTPUT LISTING DEVICE




NOTE: IF LI=TEC:
                TTY KEY          ACTION
                -------          ------
                   P             PRINT THE SCREEN ON LPT OR TTY
                  0-9            AUTO-PAGING WITH DELAY PROPORTIONAL TO NUMBER
                 OTHER           STOP AUTO-PAGING

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-38. The LOOK Mode FILE USAGE Page.

When the command RUN and XMIT is initiated from the LOOK page, the ASCII file indicated will be output to the device specified.

If TEC: has been listed as an output device, the <u>first</u> page of the file will be displayed. Depress the XMIT key or press one of the 0-9 TTY keys as described in para. 5.7.2 to advance pages. <u>Any</u> page being displayed on the data terminal can be output to the line printer (if it is active); otherwise, to the Teletype by depressing the Teletype P key.

The monitor in use is automatically entered after the last page of the ASCII file has been output.

5.6.7 DISPLAY

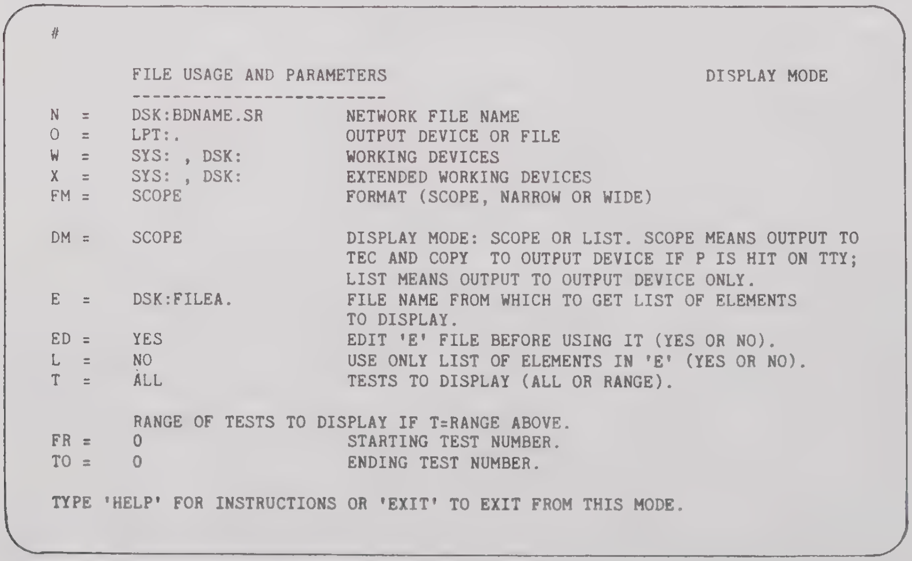
The Utility Program DISPLAY may be run <u>after</u> completing successful PREPARE Mode G or I Mode Runs. The program will output all or selected segments of the Good Board Simulation Values (PRETTY) only. The program can be entered from the CAPS or APG Monitor using any of the various approaches described in Section 3. For example,

```
# DIS   and XMIT
```

will cause the first DISPLAY page to be displayed (see Figure 5-39). The page is titled FILE USAGE AND PARAMETERS, and is divided into three sections. The N,O,W, FM, and DM options are general purpose FILE USAGE options that apply at all times. The E,ED, and L options apply only to a special ASCII file that can be prepared with EDIT. The file contains a list of those ICs and/or EXT names that are to be included in the PRETTY listing being output. The T,FR, and TO options apply to the range of test steps that are to be included in the listing. Each option may be set-up as follows:

1. <u>NETWORK FILE NAME (N=)</u>. List the network file name (.SR file) and device location of the board that has been run through at least the G or I Modes of PREPARE. For example,

   ```
   N=DSK2:SAMPLE.SR
   ```



```
#

        FILE USAGE AND PARAMETERS                               DISPLAY MODE
        -------------------------
N  =    DSK:BDNAME.SR           NETWORK FILE NAME
O  =    LPT:.                   OUTPUT DEVICE OR FILE
W  =    SYS: , DSK:             WORKING DEVICES
X  =    SYS: , DSK:             EXTENDED WORKING DEVICES
FM =    SCOPE                   FORMAT (SCOPE, NARROW OR WIDE)

DM =    SCOPE                   DISPLAY MODE: SCOPE OR LIST. SCOPE MEANS OUTPUT TO
                                TEC AND COPY  TO OUTPUT DEVICE IF P IS HIT ON TTY;
                                LIST MEANS OUTPUT TO OUTPUT DEVICE ONLY.
E  =    DSK:FILEA.              FILE NAME FROM WHICH TO GET LIST OF ELEMENTS
                                TO DISPLAY.
ED =    YES                     EDIT 'E' FILE BEFORE USING IT (YES OR NO).
L  =    NO                      USE ONLY LIST OF ELEMENTS IN 'E' (YES OR NO).
T  =    ALL                     TESTS TO DISPLAY (ALL OR RANGE).

        RANGE OF TESTS TO DISPLAY IF T=RANGE ABOVE.
FR =    0                       STARTING TEST NUMBER.
TO =    0                       ENDING TEST NUMBER.

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-39. The DISPLAY Mode FILE USAGE AND PARAMETERS Page.

lists the file for SAMPLE.SR located on DSK2:. If a device is not specified, DSK: is assumed.

2. OUTPUT DEVICE OR FILE (O=). List the output device that is to be used. The following devices may be listed:

O=LPT:	Outputs PRETTY to the line printer.
O=TEC:	Outputs PRETTY to the TEC display.
O=TTY:	Outputs PRETTY to the Teletype.
O=DSK3:SAMPLE.PO	Outputs PRETTY to DSK3: under the name SAMPLE.PO. Files can be stored on any disk device. The file will be saved under the name and extension specified. The .PO extension does not conflict with CAPS or APG generated files and may be used, if desired.

3. WORKING DEVICES (W=) EXTENDED WORKING DEVICES (x=). List the same devices that were specified during the PREPARE Mode run for the board list. CAPS or APG .NM, .ST, and .GV files are required in order to run PRETTY, and the devices that contain the files must be specified at this time. For example, if DSK2:,DSK3: were specified for W= and DSK4:,DSK5 for X= in PREPARE Mode, then these devices would contain the needed files for DISPLAY. Under these conditions the W= option and X= options would be listed as:

```
W=DSK2:,DSK3:
X=DSK4:,DSK5:
```

NOTE

The W= and X= specifications once made for any mode (for example the PREPARE Mode) are automatically changed in all applicable modes.

4. FORMAT (FM=). This option changes the character width and length of the PRETTY listing in the following manner:

Command	Maximum Width	Maximum Length
SCOPE	72 characters	24 lines
NARROW	72 characters	60 lines
WIDE	128 characters	60 lines

If TEC: display outputs are desired, always specify SCOPE.

```
FM=SCOPE
```

If Teletype data outputs (or line printer with 8½ x 11" paper) are desired, use NARROW.

```
FM=NARROW
```

If wide paper line printer data outputs are desired, either NARROW or WIDE will work, however, WIDE is recommended for the most efficient utilization of printer paper.

```
FM=WIDE
```

5. DISPLAY MODE (DM=). Either SCOPE or LIST may be specified.

SCOPE	means output data to the TEC: display and then copy the displayed data onto the output device listed under O= if the Teletype P key is depressed.
LIST	means output data to the output device listed only.

6. FILE NAME (E=). This option specifies the ASCII file name and device that contains the list of elements (ICs and externals) to be displayed. An ASCII source file can be prepared with EDIT and saved on the device indicated. If the L= option is turned ON (L=YES), the list of elements specified in that file will be the only elements that are listed in the Simulation Results. Methods of preparing the file are described in step 7.

7. EDIT E= FILE (ED=). If ED=YES, EDIT is entered automatically prior to running DISPLAY, and the file name specified by the E= option loaded into core for EDIT, if available; otherwise, a new file is created at this time. If ED=NO, and a command file exists under the name specified by the E= option, commands in that file will be used (if the L= option is YES), however, EDIT is not entered.

The file format is as follows:

```
IC2,IC4,IC33;
```

Where the devices listed will be the only ones included in the Simulation Results Listing. Individual IC Pins may also be specified. For example, the command line

```
IC2.1,IC4.3,IC33;
```

will result in a listing of the pins specified for IC2 and IC4 only.

NOTE

> The desired set of DISPLAY OPTIONS (page 2 of DISPLAY) must also be ON for the correct listing to occur when DISPLAY is run

EXTERNAL INPUTS and OUTPUTS may also be specified in the file. For example, the command line

```
# 25,#30,#31;
```

will result in a listing of the EXTERNALs listed only. Commas must separate each entry and the command line must be terminated with a semi-colon (;). The # sign is required only if names and D/S numbers are not identical (for example, 5 EXTIN 4;).

Starting and ending test numbers (FR= and TO=) can be included in the EDIT file. These options function in the same manner as the DISPLAY Mode options described in step 10.

Once the file has been completed and is in memory, depressing X and XMIT keys will cause DISPLAY to be entered and run. After outputting the Simulation Results, an automatic return to EDIT will occur so that modifications to the file can be made, if desired. E XMIT (or CTRL C) can be used to enter the CAPS or APG Monitor from EDIT.

Up to 25 devices (IC pins or Externals) can be specified in a file.

8. USE ONLY ELEMENTS IN FILE (L=). When L=YES, the list of elements in the file specified by the E= option (and no others) will be used. If L=NO, the file will not be used. Set L=NO whenever it is desired to output a complete listing of all ICs and external names in PRETTY.

9. RANGE OF TESTS (T=). If T=RANGE, then values for the range of test steps specified by the FR= and TO= options will be listed. If an EDIT file has been created which contains FR= and TO= specifications these values will override the DISPLAY Mode Monitor Page values.

If T=ALL then all test steps will be included in the listing, unless the EDIT file contains FR= and TO= options, in which case those test steps specified will be listed and the T=ALL option overriden.

10. STARTING AND ENDING TEST NUMBERS (FR=,TO=). The starting and ending test number options specify the range of test steps to be included from the Good Board Simulation Results. For example,

```
FR=69
TO=71
```

will result in a listing of test steps 60 through 71, while

```
FR=4
TO=11
```

will result in a listing of test steps 0 through 11. Note that in each case, the lower limit of tests (FR=) resulted in an actual listing of tests that were greater than the limits specified. This occurs because test steps are actually output in even multiples of ten (10,20,30,etc.). The lower limit therefore specifies the lowest group of ten test steps to be output. The higher limit will truncate the output of test steps at the number specified.

After establishing the desired options for page 1, depress the XMIT key again for page two of DISPLAY. The second page is identical to the DISPLAY OPTIONS page shown in Figure 5-20 with the exception that single STEP EXPANSION options are not included. Single step expansion listings may be output with DISPLAY when a complete run is made (T=ALL, L=NO, ED=NO). The options listed on the DISPLAY OPTIONS page are described in Para. 5.3.8, and can limit the major segments of the Simulation Results listing that is to be output (list EXTERNAL INPUTS, IC INPUTS, IC OUTPUTS, etc.) or change the 1, 0, X format used.

Once both pages of DISPLAY have been set to the desired values, the RUN command can be issued to run the program.

```
# RUN and XMIT
```

The CAPS or APG Monitor is again entered after the desired listing has been output if ED=NO. If ED=YES, a return to EDIT occurs after outputting data so that the display for additional points can be requested.

5.6.8 CLEANUP

The Utility Program CLEANUP can be used to delete those CAPS and APG test, listing, and temporary files that were created during PREPARE and GENERATE mode runs. The program also deletes selected listing and test files created during TRANSLATE and DIAGNOSE mode runs. Source files created by the user including the basic test program (.TP files), network description (.SR file), or probe starting point (.US) file and others are not deleted by CLEANUP. The program deletes CAPS files of the same network name only, thus making it possible to selectively delete those files that are no longer required while saving files that are still needed.

The program can be entered from the CAPS, APG, or TEST Monitor using any of the various approaches described in Section 3. For example,

```
# CLE  and XMIT
      or
# DEL  and XMIT
```

will cause the first and only CLEANUP page to be displayed (see Figure 5-40). This is a FILE USAGE page that may be set-up as follows:

1. NETWORK FILE (N=). List the network file name (the name assigned to the .SR file)that is to be deleted. For example,

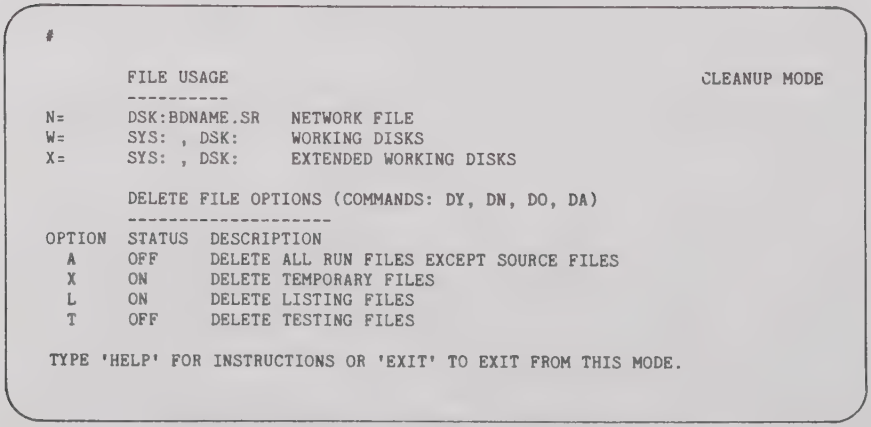
   N=DSK2:SAMPLE.SR

   means that CAPS or APG files with the name SAMPLE and the extensions listed in step 3 are to be deleted. Files with other names will not be deleted. Note that although the .SR file is listed, it will not be one of the files to be deleted as it is a source file that was not created by CAPS or APG. If a device is not specified, DSK: is assumed. If an extension is not specified, the .SR extension is assumed.

2. WORKING DISKS (W=), EXTENDED WORKING DISKS (X=). List the devices that contain the files to be deleted. Table 5-11 describes the CAPS or APG listing, test, and temporary files created during PREPARE Mode (refer to para. 5.3.2.6).

   NOTE

   When W= and X= have been specified for any other mode, the change is automatically made, and applies to all applicable modes.



```
#

        FILE USAGE                                          CLEANUP MODE
        ----------
N=      DSK:BDNAME.SR   NETWORK FILE
W=      SYS: , DSK:     WORKING DISKS
X=      SYS: , DSK:     EXTENDED WORKING DISKS

        DELETE FILE OPTIONS (COMMANDS: DY, DN, DO, DA)
        --------------------
OPTION  STATUS  DESCRIPTION
  A     OFF     DELETE ALL RUN FILES EXCEPT SOURCE FILES
  X     ON      DELETE TEMPORARY FILES
  L     ON      DELETE LISTING FILES
  T     OFF     DELETE TESTING FILES

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 5-40. The CLEANUP Mode FILE USAGE Page.

3. DELETE FILES OPTIONS (A,X,L,T). Each of the options will delete files of the same name as the file listed by N= in the following manner:

A ON	This option deletes all CAPS, TEST, and APG non-user entered files with the name specified by N=. After running CLEANUP with A ON, a new PREPARE Mode run (or TEST Monitor TRANSLATE Mode run) will have to be completed before the board named can be tested in the DIAGNOSE Mode.
X ON	Delete CAPS or APG Temporary files only. Files with the name specified by N= are deleted. Temporary file deletions do not prevent a board from being tested again in the DIAGNOSE Mode.
L ON	Delete CAPS or APG listing files only. Files with the name specified by N= are deleted. Listing file deletions do not prevent a board from being tested again in the DIAGNOSE Mode.
T ON	Delete CAPS or APG testing files only. Files with the name specified by N= are deleted.

When the command RUN and XMIT is initiated from the CLEANUP Page, the files specified will be deleted and the monitor in use (CAPS, APG, or TEST) entered.

5.7 SPECIAL FEATURES

5.7.1 General

The special features described in the following paragraphs may be used when in the PREPARE Mode to:

1. Automatically advance pages displayed on the TEC data terminal during a PREPARE or GENERATE mode run (refer to Para. 5.7.2.).

2. Print pages displayed on the TEC data terminal on the line printer if it is active; otherwise, on the Teletype (refer to Para. 5.7.3).

3. Suppress data outputs or perform other functions during a PREPARE Mode run through the use of special interactive Teletype keys (refer to Para. 5.7.4).

5.7.2 Automatically Advancing TEC Terminal Pages

TEC Data Terminal pages can be advanced automatically during PREPARE and GENERATE mode runs through use of a special TEC Handler that provides the following unique advantages:

1. Data pages displayed on the TEC Terminal may be advanced by depressing the XMIT key, as before.

2. Any Teletype key except numbers 0-9, and the Interactive Teletype keys listed in Para. 5.7.4 can be depressed to advance data pages in exactly the same manner as is accomplished with the XMIT key.

3. When a number from 0-9 is depressed on the Teletype keyboard, pages are automatically advanced at a rate determined by the key that has been depressed. The rate is in 1/2 second multiples of the number depressed; that is, a "1" will cause pages to advance at the rate of one page each ½ second, etc. When a 0 is typed, page advance is very rapid. (The use of 0 is practical only when there are many pages.) This automatic advance may be halted at the current page being viewed by depressing any letter not excluded in step 2. If a different number is typed while in the "automatic" mode, the rate dictated by that number will be immediately assumed.

5.7.3 Printing Pages Displayed On TEC

Individual pages that are displayed on the TEC data terminal may be output to the Line Printer by depressing the Teletype P key while the page is displayed.

NOTE

> If the Line Printer is inactive when the P key is depressed, test system operations will halt and the system will not respond to CTRL C. Turn the Line Printer ON and then execute a CTRL C to return to the CAPS or APG Monitor.

5.7.4 Interactive Teletype Keys

The following Teletype keys may be used to accomplish the actions indicated:

Key	Function
CTRL C	Terminates the PREPARE Mode operation in progress and returns to the CAPS or APG Monitor.
CTRL O	Terminates listings being output on the Teletype only and proceeds with the next CAPS or APG operation. Use the letter O.
CTRL Q	Terminates (Quits) PREPARE Mode listings being output to any device and proceeds with the next CAPS or APG operation. The command is ignored if data is not being output.
CTRL S	Displays the name of the current CAPS or APG Module being run.
CTRL X	Terminates the AFD Simulation run in progress and returns control to the CAPS or APG Monitor. A RESTART will occur at the last check point taken when the F Mode is again run if the RESTART option is also ON (RS=YES).
P	Outputs the page displayed to the Line Printer. Make sure that the Line Printer is active before depressing the Teletype P key or the test system will hang-up. If the test system does hang-up, turn the Line Printer ON and then execute a CTRL C to return to the CAPS or APG Monitor.

Diagnose Mode-Section 6

6.1 INTRODUCTION

6.1.1 General

This section contains a description on the various diagnostic techniques and tools available to a user in the DIAGNOSE Mode. It includes a description on the overall functional flow used in the four sub-modes of DIAGNOSE (CONTROL BY CAPS SECTION, GUIDED PROBE, AUTOMATIC (AFL), and MANUAL), together with methods of setting DIAGNOSE Mode options, interpreting diagnostic messages and testing boards.

6.1.2 Sub-Mode Memory Requirements

The enhanced capabilities provided by CAPS-X require a minimum of 32K memory to run DIAGNOSE Mode.

In addition, the correct number of PREPARE Mode Sub-modes must have been run previously in order to generate the needed data base files required to run the CAPS/APG DIAGNOSE Mode with either the PROBE and/or AFL enabled (refer to Para. 5.2). When using the MANUAL Sub-mode of DIAGNOSE, a translated binary version of the test program is required. If the correct data base files have not been generated, a message of the form

```
CANT FIND DEV:NAME.EX
```

will occur when the RUN command is issued, followed by an immediate return to the Monitor top level. The PREPARE (or TRANSLATE) Mode may then be entered and the necessary runs begun to create the needed files.

6.2 DIAGNOSTIC TECHNIQUES AVAILABLE

6.2.1 General

The various diagnostic Sub-modes, options for their usage, guided probes/clips, Diagnostic Resolution Module (DRM), control panel, and diagnostic message formats are all designed to optimize the throughput, accuracy, and flexibility of your GenRad test system when testing and diagnosing circuit board failures. Two of th Sub-modes provided (GUIDED PROBE and AFL) allow access to the unique features of GenRad's computer-guided probe/clip and FINDS (Fault Identifying Nodal Diagnostic Software) package. AFL (Automatic Fault Location) may also be used by itself in order to determine board-failure sources without operator intervention, when desired. The MANUAL Sub-mode provides board testing and trouble-shooting techniques through use of the various test system controls and displays (see Figure 6-1).

The DIAGNOSIS CONTROL BY CAPS SECTION Sub-Mode allows SETUP Mode options and DIAGNOSE Sub-modes and option pages to be set up by individual Multiple CAPS sections. (See 179X Test Programming Manual, Appendix C, for the description of Multiple CAPS sections.)

6.2.2 Guided PROBE/Clip Sub-Mode

The Guided Probe/Clip is a powerful diagnostic tool that can track board failures back through a circuit from the board's external pins, from user specified starting (or jumping) points, or from internal board starting points that have been passed to the probe/clip through the probe link. Internal starting points, passed through the probe link, may be generated because of a Quick Fault Dictionary Lookup (QUICK LOOKUP), or because of an initial/final AFL run. Other probe options allow selection of a BED-OF-NAILS feature that automatically reads the state of nodal sensor pins (NAILS) thus speeding up the probing sequence, and the FINDS package. FINDS may be used after completing a probe/clip run to further isolate nodal faults (such as short or open conditions between and IC pin and power or ground, or between adjacent IC pins, faulty IC's, etc.), when required.

The Guided Probe/Clip directs an operator to place the probe (or clip) on specific circuit points (or IC's). The probe/clip software then analyzes the data obtained at these points in order to isolate the failure to a particular circuit point or node. The probe/clip software instructs an operator as to the points to probe (or clip) via the TEC display. The operator then places the probe (or clip) on the point indicated and depresses the PROBE button on the system control panel (or the optional foot-switch). The control panel ERROR RECOVERY button and the TEC keyboard allow an operator to recover from a mis-probe/clip and to direct the probe/clip in a more explicit manner.

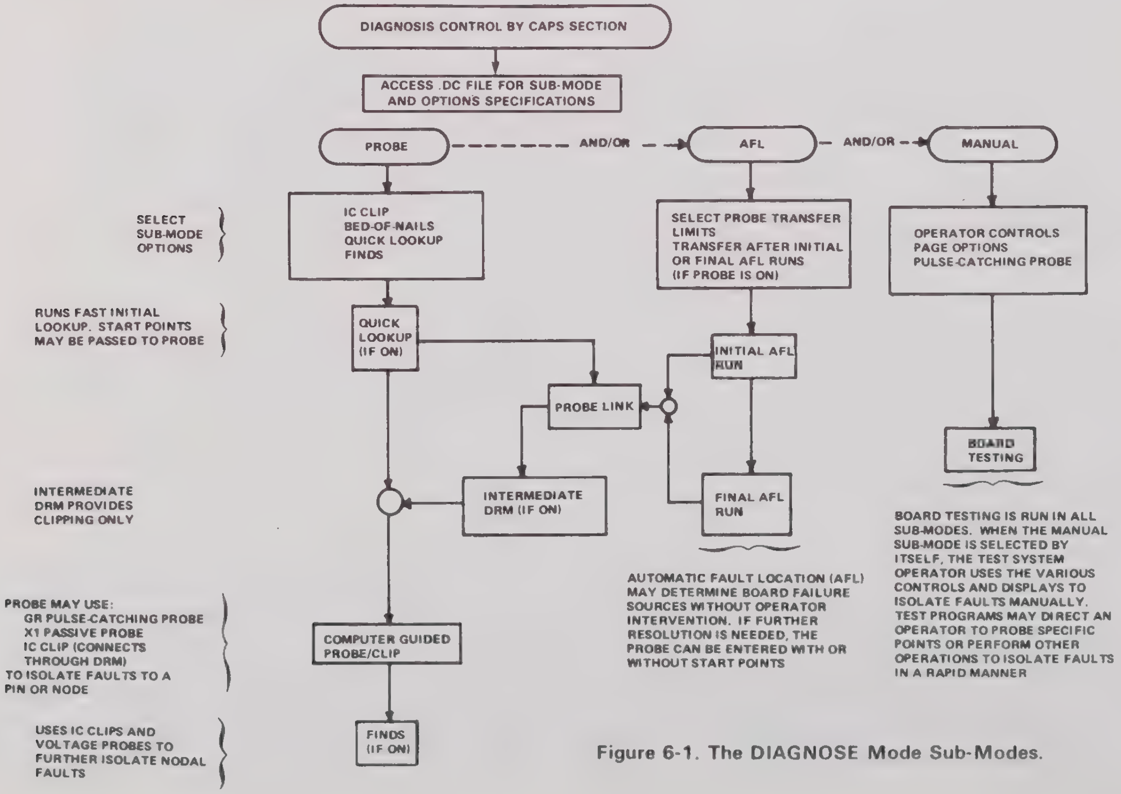

DIAGNOSIS CONTROL BY CAPS SECTION
ACCESS .DC FILE FOR SUB-MODE AND OPTIONS SPECIFICATIONS
PROBE
AND/OR
AFL
AND/OR
MANUAL
SELECT SUB-MODE OPTIONS
IC CLIP
BED-OF-NAILS
QUICK LOOKUP
FINDS
SELECT PROBE TRANSFER LIMITS
TRANSFER AFTER INITIAL OR FINAL AFL RUNS (IF PROBE IS ON)
OPERATOR CONTROLS
PAGE OPTIONS
PULSE-CATCHING PROBE
RUNS FAST INITIAL LOOKUP. START POINTS MAY BE PASSED TO PROBE
QUICK LOOKUP (IF ON)
INITIAL AFL RUN
PROBE LINK
BOARD TESTING
INTERMEDIATE DRM PROVIDES CLIPPING ONLY
INTERMEDIATE DRM (IF ON)
FINAL AFL RUN
BOARD TESTING IS RUN IN ALL SUB-MODES. WHEN THE MANUAL SUB-MODE IS SELECTED BY ITSELF, THE TEST SYSTEM OPERATOR USES THE VARIOUS CONTROLS AND DISPLAYS TO ISOLATE FAULTS MANUALLY. TEST PROGRAMS MAY DIRECT AN OPERATOR TO PROBE SPECIFIC POINTS OR PERFORM OTHER OPERATIONS TO ISOLATE FAULTS IN A RAPID MANNER
AUTOMATIC FAULT LOCATION (AFL) MAY DETERMINE BOARD FAILURE SOURCES WITHOUT OPERATOR INTERVENTION. IF FURTHER RESOLUTION IS NEEDED, THE PROBE CAN BE ENTERED WITH OR WITHOUT START POINTS
PROBE MAY USE:
GR PULSE-CATCHING PROBE
X1 PASSIVE PROBE
IC CLIP (CONNECTS THROUGH DRM)
TO ISOLATE FAULTS TO A PIN OR NODE
COMPUTER GUIDED PROBE/CLIP
USES IC CLIPS AND VOLTAGE PROBES TO FURTHER ISOLATE NODAL FAULTS
FINDS (IF ON)

Figure 6-1. The DIAGNOSE Mode Sub-Modes.

Starting point (.US) files, jumping point (.UJ) and conditional-connectivity (.UC) files prepared by a user enable the test programmer to alter the probing sequence. Starting points instruct the guided probe to start probing at user-suspected failure points internal to the board. Jumping points instruct the probe to jump to user-specified points if user-defined internal points fail during normal probing.

When the PROBE sub-mode is enabled, only test program commands inserted between the BEGIN CAPS and FINISH CAPS commands are executed. Test program commands outside of this area are ignored. However, on the first pass, all commands up to the FINISH CAPS command of the failing section are executed.

6.2.2.1 Probe/Clips Available

Either a standard X-1 passive probe (that can record good and bad logic levels) or a GenRad pulse-catching probe (that can also record illegal or missing pulses, pulse transitions, and oscillations) may be used as the computer-guided (smart) probe. In addition, a 16 (or 24) pin IC Clip can be used in place of the X-1 probe, but does not provide any pulse-catching capability. The IC Clip attaches to DRM CLIP B, and is therefore only available if the optional DRM is supplied.

NOTE

The IC Clip when used, must always connect to DRM CLIP B (never CLIP A).

X-1 Passive Probe

The X-1 Passive Probe records only logic states. The probe may be connected to one of the AUXiliary BNC jacks on the test system support shelf, and then wired to a dedicated Sensor pin using the procedures contained in Para. 2.2.2.

GenRad Pulse-Catching Probe

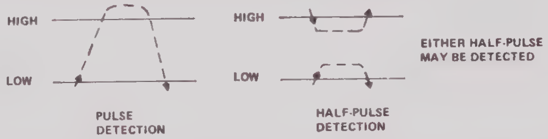
The GenRad pulse-catching logic probe is an integrated hardware/software package that detects bad logic levels, illegal or missing pulses, and transitions between voltage levels. The hardware portion consists of a hand-held logic probe, connecting cable, and probe control board. The hand-held probe connects to the 14-pin LOGIC PROBE Jack. This jack may be located at the rear of the test-system support shelf or on the side of the shelf assembly (depending on the test-system configuration in use). The probe control board resides in the test-system I/O Box. The software matches basic transition and final level data obtained from the hardware probe with simulated data. Mis-matches between the two are used for a probe diagnosis unless directed otherwise by an operator through use of a special user pulse-discrimination (.UD) file. The pulse-catching logic probe will detect the following types of errors:

1. A bad logic level. The probed point may either be stuck or active and in one of three states (High, Low, or Mid-range).

2. Oscillations between level states.

3. An illegal or missing pulse. A pulse that occurs when it should not is considered illegal, while a pulse that does not occur when it should is considered to be a missing pulse.

4. A half-pulse. Four possible (directional) signal transitions occur across threshold boundaries when a complete pulse has been detected. A half-pulse is considered to be a pair of transitions as indicated by the following two examples:



The pulse-catching logic probe provides the capability of detecting errors by comparing the UUT voltage level at the point being probed with two threshold reference voltages. The threshold voltages are selectable over a nominal ±30V range to accomodate mixed logic families on the same board, and are established by the SETUP Mode EAL and EBL options in 1792 test systems and the ETL option in 1795 test systems. Threshold levels may also be changed automatically by the probe software as different IC family types (TTL, HTTL, etc.) are probed. Methods of selecting the probe threshold levels together with the logic family threshold levels are described in the System Operating Instructions.

The probe hardware does not attempt to preserve a total waveform history during a test step, but instead, supplies key parameters only (final voltage levels and transitional data) to the probe software. The probe software uses the data supplied to generate the simplest possible waveform model that can be constructed (which may not be an exact algorithmic representation of the original waveform being probed). The waveform model is then compared to the simulator produced waveform that is generated from nodal data obtained from a PREPARE Mode Analysis of Good Network (AGN) simulation run. Mis-matches between the two waveform models can either be ignored by the probe software (if directed by a user through use of a special User Pulse Discrimination File) or used by the software for a probe diagnosis.

IC Clip

The IC Clip consists of a 16, 24 or 40-pin IC Clip, connecting cable, and terminating connector. When using the computer-guided probe/clip, the terminating connector must always be attached to DRM CLIP B (never to CLIP A). These Clips are the same as those supplied for use with FINDS (refer to para 6.2.3.1).

The 16 pin clip is always used to clip ICs having 16 pins or less. The 24 pin clip handles ICs having greater than 16 pins. Pin 1 of each clip must be placed on pin 1 of the IC unless 14 pin ICs are being clipped. With 14 pin ICs, clip pins 1 or 2 may be placed on IC pin 1.

In order to use the IC Clip with the guided probe/clip software, the DIAGNOSE Mode GUIDED PROBE OPTIONS Page I Option (USE IC CLIP) must be ON. In addition, the SETUP Mode, SYSTEM CONFIGURATION Page DR= Option must be set to DR=1 or 2. Once these software switches have been set, the probe/clip software will ask the operator to CLIP IC's rather then probe them. The software stores information covering the entire IC in the probe data base. The points needed are then used by the probe software to speed-up the probing sequence. If an IC cannot be clipped, it is only necessary to depress the XMIT Button in order to receive point-to-point probing instructions. In addition, the software remembers that an IC is not to be clipped and will not ask that the IC be clipped again during the same test sequence. However, if the START Button is again depressed to start a new test sequence, the software will again request that the IC be clipped (unless the XMIT Button is again depressed). This reduces probing time during a normal sequence.

6.2.2.2 Bed-of-Nails Feature

The Bed-of-Nails feature is enabled whenever the DIAGNOSE Mode GUIDED PROBE OPTIONS Page N Option is turned ON. This feature eliminates manual probing of nodes that are connected to external Sensor pins (NAILS) during the normal probing sequence. It is ideally suited for circuit nodes that have a high degree of fan-out (perhaps a group of inputs are all being driven by the same output pin which has a Sensor pin (NAIL) attached to it).

The following points should be observed when using the Bed-of-Nails feature:

1. Nodal checkpointing is not accomplished if the Bed-of-Nails feature is ON.

2. If two or more Sensor pins are connected to the same node, the lowest numbered Sensor pin will be used as the NAIL.

6.2.2.3 Quick Dictionary Lookup Feature

The Quick Dictionary Lookup Feature is enabled whenever the DIAGNOSE Mode GUIDED PROBE OPTIONS Page Q Option is turned ON. The feature quickly passes starting points (obtained from a quick fault dictionary lookup) through the probe link to the computer-guided probe if AFL is not enabled. This feature does not work if the AFL Sub-mode of DIAGNOSE is turned ON. The Quick Dictionary Lookup method may prove significantly faster then diagnosing faults with AFL enabled if a user does not intend to perform a complete AFL run; but instead, wishes to pass starting points to the probe from an initial AFL Lookup only.

The data base files required to run the Quick Lookup feature are made when the PREPARE Mode Q Sub-mode is run (refer to Para. 5.2.8).

6.2.3 FINDS

6.2.3.1 General

FINDS (Fault Identifying Nodal Diagnostic Software),in conjunction with a DRM (Diagnostic Resolution Module), allows the software to further diagnose nodal failures that the guided probe/clip cannot resolve. A new DRM (hereafter referred to as DRM II) is available as of the release of CAPS System X, Release 01. The new software supports either DRM II or the original DRM (hereafter called DRM I). Both units, and the accompanying software, expand testing capability to include detection of:

1. Shorts/opens between an IC pin and a power or ground pin.
2. Shorts/opens between adjacent IC pins.
3. Faulty ICs.
4. Shorts between tracks on an etched board.

FINDS testing may be performed when the computer-guided probe/clip has completed its diagnostic run and has detected a faulty node. The probe software automatically links to FINDS when the XMIT button is pressed if the following conditions are met:

1. The GUIDED PROBE nodal diagnosis is displayed on the TEC terminal.
2. The GUIDED PROBE OPTIONS page F (FINDS) Option is enabled.
3. The SETUP mode SYSTEM CONFIGURATION Page DR (DRM Option is set to DR=1 or DR=2.
4. If their use is desired, the Track Short Detection (T) and Low Sensitivity (L) Options must be specified. These options are described in para 6.3.

For suspected broken connections, FINDS requests the operator to probe/clip each side of the node simultaneously to confirm the broken connection. If it is not confirmed a probable misprobe is diagnosed and the board should be retested. For faults other than broken connections FINDS checks first to see whether a stuck node is shorted to power or ground. FINDS then checks for shorts from the bad node to all pins on the nodal IC's. The test is referred to as the adjacent pin short test. Diagnosing both types of shorts is accomplished by checking the continuity between the nodal IC pin and either power or ground or all other IC pins on the same IC. If continuity does exist but should not then the fault is diagnosed as a short. If continuity that is expected is not found then the fault is diagnosed as an open. Next, if an adjacent pin short fault has not been found, the system checks each IC output driving the faulty node to verify that the IC is not faulty. To verify that the IC is not faulty, voltage magnitude and direction is measured at the IC output pins. If the output of the IC pin is bad-low (node should be high) and the pin is not sourcing current then the IC is bad. Note that the DRM measures voltage not current. For systems equipped with DRM II, voltages are also checked on the IC inputs.

The last operation performed by FINDS is to check the continuity between tracks on the etch board. This procedure is accomplished under operator control. If continuity exists where it should not, then the fault is diagnosed as a short between the checked tracks. This test is referred to as the track short test.

6.2.3.2 FINDS Probes/CLips (DRM I)

The DRM I is equipped with two probes and a set of chip clips. The set of chip clips is composed of two 16 pin chip clips and two 24 pin chip clips. The clips are interchangable, but the DRM can only support two clips at a time. The DRM is configured to handle two 24 pin clips, two 16 pin clips, or a combination of one 16 pin clip and one 24 pin clip. The clip that is designated as the B clip is a "smart" clip because the DRM has the capability to detect contactivity between it and the IC package.

6.2.3.3 FINDS Probes/Clips (DRM II)

The DRM II offers enhanced probing/clipping capabilities. Included are 2 ohmmeter probes, a microvoltmeter probe, and a set of chip clips. This set includes a new 40-pin clip and a new 20-pin clip as well as 16 and 24-pin clips of the old type. All may be used as "smart" clips.

6.2.4 AFL Sub-Mode

Automatic Fault Location (AFL) is a technique featuring a two step process for automatically determining the source of a board failure without operator intervention. The first step, a fault dictionary lookup, isolates the possible fault to one of a small number. The second step, a simulation of the possible failures in which the behavior of the faulty board is compared to the simulated values, further isolates the possible defects causing board failure.

In order to utilize the best features of both steps an "AFL to Probe" link is included. Through the link possible points of failure internal to the circuit, as determined by AFL, are passed to the Guided Probe. The Guided Probe then directs the operator to probe the AFL indicated points and further proceeds to isolate the failure down to a circuit node or point. This resolution is often not possible using AFL alone.

Automatic Fault Location depends on a partial fault dictionary created in the PREPARE Mode F sub-mode. During device testing under the AFL testing technique, when a board fails, the first failing test and failing pins are recorded. The fault dictionary is then searched for this failing test to determine which faults could possibly be responsible for the failure. This is called the initial fault dictionary lookup.

AFL simulation is then invoked to simulate these faults. During the process of AFL simulation, the faults from the initial lookup are inserted into the simulation model of the UUT. A fault simulation, similar to the one performed in the fault detection simulation of PREPARE mode, is then performed. The major difference from the PREPARE mode fault simulation is that the AFL simulation compares its results with the behavior of the faulty UUT. The AFL simulation differentiates three types of comparison matching:

1. Verified - the modelled fault exactly matches the UUT's behavior.

2. Verified, but others - modelled fault's failures are a subset of the UUT's failures - indicating that the fault and one or more other faults are present on the failing UUT.

3. Probable - the modelled fault did not match the UUT's behavior exactly but was close enough to indicate that the fault is in the area of the actual problem.

6.2.4.1 AFL to Probe Link

During the AFL process, the number of possible fault classes responsible for a board's failure are narrowed from the initial fault dictionary lookup fault classes to those left after AFL simulation. After an initial lookup is performed, however, the number of fault classes under consideration may be small enough so that it is worthwhile to immediately start probing rather than continue on to AFL simulation. The user may set options such that this transfer is effected if the number of fault classes is less than a certain preset number. This AFL to Probe link causes the Guided Probe to look at one point from each fault class to determine which possible fault is actually responsible for the circuit's failure. AFL simulation may be used to narrow down the number of fault classes under consideration. After AFL simulation is performed, the fault class or classes remaining may not give sufficient resolution. Using the AFL to Probe link, the user may choose to have the Guided Probe/FINDS further isolate the cause of failure. Again, this is done by choosing one point from each possible fault class.

6.2.4.2 Intermediate DRM

Intermediate DRM checks for shorts or open conditions between all pins on the IC. It uses internal board-starting points passed from a probe quick lookup, INITIAL AFC or FINAL AFL run to help locate shorts or open conditions with the DRM and thus avoid the necessity of having to continue on to the guided probe for this type of failure.

Intermediate DRM may be entered automatically if the DIAGNOSTIC MODES A (AFL) or G (GUIDED PROBE) options have been turned ON, and the IC= option has been set to indicate the maximum number of ICs to be clipped. As with FINDS, the SETUP Mode SYSTEM CONFIGURATION page DR (DRM) option must also be set to DR=YES.

The DIAGNOSIS Mode IC= option is normally set to a value which defines the maximum number of ICs an operator wishes to clip once the Intermediate DRM is entered. If the actual number of indicted ICs exceeds this value (or IC=0), Intermediate DRM is bypassed and the starting points passed directly to the Guided Probe/Clip. If no diagnosis is determined, by Intermediate DRM, that is no illegal opens or shorts were found, the starting points are then passed to the probe. If no AFL or Quick Look starting points are transferred, then Intermediate DRM will not be entered.

When using Intermediate DRM, depressing the ERROR RECOVERY button results in a return to the first starting point (not the last IC clipped) and the user must reclip all of the ICs again. Depressing the XMIT button will cause the next IC to be clipped.

6.2.4.3 System Flow of Control

Figure 6-2 is an overview of the system flow between diagnostic techniques and program modules. The < or > diamonds indicate that at these points decisions are made based on parameters set up through the use of the monitor. The boxes represent modules and the labelled circles represent control panel buttons.

The testing process is initiated by entering the CAPS/APG system from the OS/8 monitor (by typing .R CAPS or .R APG). Once the CAPS/APG system has been entered, the "DIAGNOSE" command is used to enter the diagnosis mode of operation; at this point the monitor options and specifications may be set. The actual physical testing process is initiated via the "RUN" command. When RUN is issued, a READY TO DIAGNOSE Message appears. At this point, the START Button may be depressed to initiate the testing sequence.

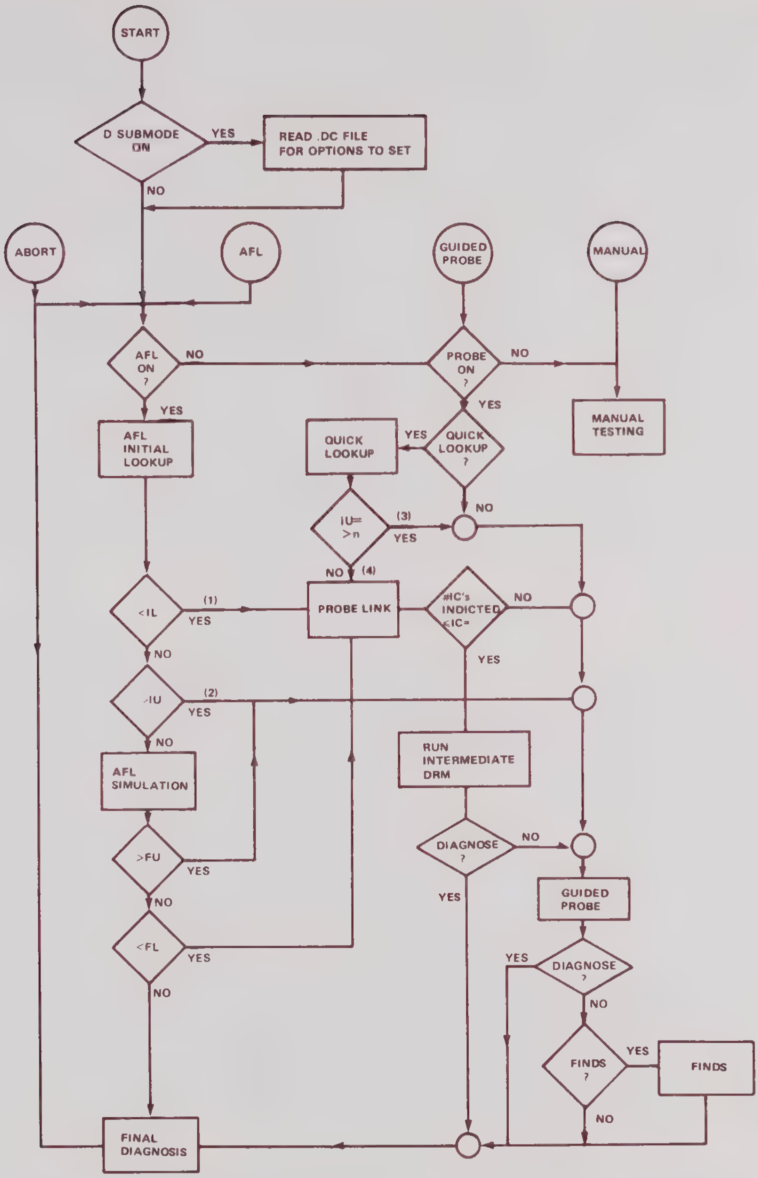

START
D SUBMODE ON
YES
READ .DC FILE FOR OPTIONS TO SET
NO
ABORT
AFL
GUIDED PROBE
MANUAL
AFL ON ?
NO
PROBE ON ?
NO
YES
MANUAL TESTING
YES
AFL INITIAL LOOKUP
QUICK LOOKUP
YES
QUICK LOOKUP ?
NO
IU= >n
(3)
YES
NO
(4)
<IL
(1)
YES
PROBE LINK
#IC's INDICTED ≤IC=
NO
NO
YES
>IU
(2)
YES
NO
AFL SIMULATION
RUN INTERMEDIATE DRM
DIAGNOSE ?
NO
>FU
YES
NO
YES
GUIDED PROBE
<FL
YES
YES
DIAGNOSE ?
NO
NO
FINDS ?
YES
FINDS
NO
FINAL DIAGNOSIS

Figure 6-2. Overall Diagnosis Mode Flow.

It can be seen that the ABORT, AFL, GUIDED PROBE or MANUAL buttons can be used to control the flow of test methods used. The START button controls the actual initiation of testing. The ABORT button is different in that it causes the disconnection of test equipment as quickly as possible. The AFL, GUIDED PROBE and MANUAL buttons each cause an ABORT followed by immediate transfer to their respective diagnostic technique if it is enabled.

If the AFL path is taken, the AFL initial lookup is first performed. Options then allow transfer to Intermediate DRM or to the Guided Probe/FINDS with (1) AFL starting points or if without (2) AFL starting points transfer to Guided Probe/Finds only. If neither of these transfers is taken, then the AFL simulation module is run. Transfers to the Guided Probe/FINDS with and without starting points are also possible here. Alternatively, a final diagnosis may be printed and new UUT testing initiated.

If the PROBE path is taken, and options are set to allow a Quick Lookup, the quick fault dictionary Lookup is performed. The DIAGNOSE Mode IU= option then allows transfer to Intermediate DRM or Guided Probe/FINDS with (4) Quick Look starting points or without (3) Quick Look starting points transfer to the Guided Probe/FINDS only. After completing a probe diagnosis, FINDS may be entered if further resolution of nodal failures is required.

6.2.5 MANUAL Sub-Mode

Board testing is performed in all sub-modes (PROBE, AFL, and MANUAL). When the MANUAL sub-mode is used by itself, the status display of the system Driver/Sensors and measured values, as well as any printouts supplied by the test program, provide the data which the operator uses to diagnose a failure. Many test program displays are possible. Using the system control panel and command monitor, the operator may run the test program in a number of modes including single step, restart on pass and/or fail, and others. These options allow the operator to investigate very difficult problems such as intermittency and dynamic malfunctions.

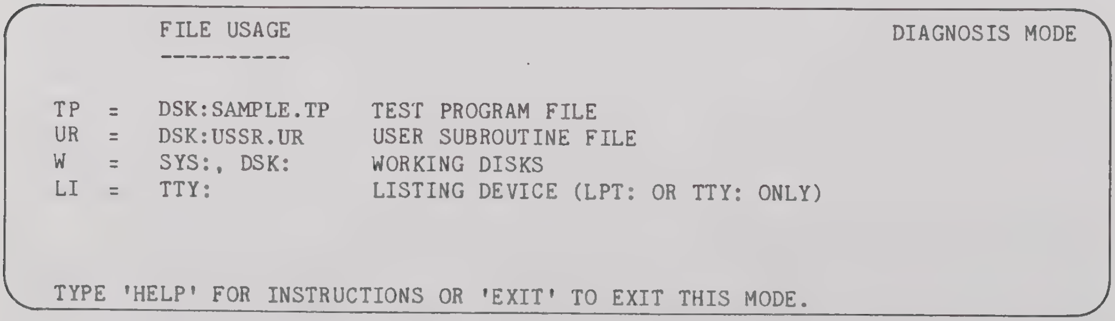
6.2.6 DIAGNOSIS CONTROL BY CAPS SECTION Sub-mode

Board testing using the Multiple CAPS technique described in Appendix C of the 179X Test Programming Manual can be optimized by using the DIAGNOSE Sub-mode most appropriate for each CAPS section. The D Sub-mode of DIAGNOSE allows each section of Multiple CAPS sections to control certain SETUP Mode options and DIAGNOSE Mode options before DIAGNOSE is run on that section. The SETUP Mode options which can be set are the threshold voltages for logic families and guided-probe operation. DIAGNOSE Mode options which can be selected include the Sub-mode (PROBE or AFL) and option pages which affect these Sub-modes. DIAGNOSE Mode option pages which apply to the MANUAL Sub-mode only are not changed by the .DC file.

The control of SETUP Mode and DIAGNOSE Mode options is accomplished by creating a .DC (DIAGNOSIS CONTROL) file for each of the individual Multiple CAPS sections. To create a .DC file for a Multiple CAPS section, set the option pages of SETUP Mode and DIAGNOSE Mode as required for that CAPS section. From the DIAGNOSE Mode enter the command

```
SAVEDC sectionname<XMIT>
```

using the same sectionname as used in the BEGIN CAPS 'sectionname';statement of the CAPS section. The .DC files should be written on the second Working Disk.



```
          FILE USAGE                                              DIAGNOSIS MODE
          ----------

   TP  =  DSK:SAMPLE.TP    TEST PROGRAM FILE
   UR  =  DSK:USSR.UR      USER SUBROUTINE FILE
   W   =  SYS:, DSK:       WORKING DISKS
   LI  =  TTY:             LISTING DEVICE (LPT: OR TTY: ONLY)



   TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 6-3. DIAGNOSE Mode FILE USAGE Page.

When the program branches to each Multiple CAPS section during a DIAGNOSE Mode run, the .DC file associated with the section is read and the SETUP Mode and DIAGNOSE Mode options are set as specified in the .DC file.

NOTE

> A .DC file must be created for each Multiple CAPS section of a board which will be tested using the D Sub-mode of DIAGNOSE. Failure to create a .DC file for each section will result in an error, terminating the DIAGNOSE run.

6.3 DIAGNOSE MODE OPTIONS

6.3.1 General

All of the optional commands necessary to allow efficient operation in the DIAGNOSE Mode are contained on eight separate pages within the mode. Methods of changing page options, advancing pages, etc. are described in Section 3. The following paragraphs describe the options available on each page and briefly define their use.

6.3.2 FILE USAGE Page

The FILE USAGE Page is the first page displayed when the DIAGNOSE Mode is entered from the monitor top level. The page contains file usage options (see Figure 6-3). The commands listed in the following paragraphs may be issued from this page.

Test Program File (TP=)

This is the name assigned to the basic source level test program (.TP File) whose binary equivalent (.TA for 1792A/B's, .TD for 1792C/D's, .TL for 1795's) is to be used for testing. The .TP file must have been translated into a test binary file either through use of the TRANSLATE MODE (for MANUAL sub-mode operation) or by the D sub-mode of PREPARE for PROBE or AFL sub-mode operation.

If a device is not listed, the DSK: device is assumed. A file labeled NAMEA whose binary equivalent file resides on DSK: could therefore be specified as:

```
TP=NAMEA
```

User Subroutine File (UR=)

This is the name of any user generated OS/8 file that will be invoked from the test program through use of an EXECUTE statement. The file must be prepared in PAL8 assembly language and saved through use of the OS/8 SAVE command using the procedures contained in Appendix B to the programming Instructions (1792-0576). Files may be located on any file structured device. If a device is not specified, DSK: is assumed. If an extension is not specified, the .UR extension is assumed.

Working Disks (W=)

The Data base files prepared by the PREPARE Mode that are needed for testing in the DIAGNOSE Mode are specified by the W= option. Normally, the option is set to reflect the devices used during the PREPARE Mode run that created the files. For example, if DSK2:, DSK3: were specified for the PREPARE Mode run, then these devices would contain the needed DIAGNOSE Mode data base files. Under such conditions, the W= option would be listed as:

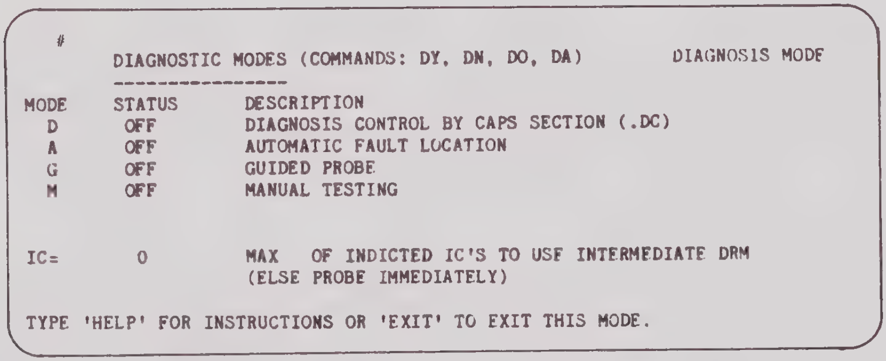
W=DSK2:,DSK3:

Listing Device (LI=)

This is the output device on which all "hard" copy (as opposed to "soft" TEC Terminal displays) will be printed. The line printer (LPT:) and the Teletype (TTY:) are the only devices supported for this specification.

6.3.3 DIAGNOSTIC MODES Page

The DIAGNOSTIC MODES Page is the second page displayed (see Figure 6-4). Options on the page allow selection of the DIAGNOSE Mode sub-mode that is to be run (PROBE, AFL, or MANUAL). If all options are turned ON at once, the AFL sub-mode will be entered. The following sub-mode selections are possible.



```
   #
          DIAGNOSTIC MODES (COMMANDS: DY, DN, DO, DA)       DIAGNOSIS MODE
          -----------------
MODE      STATUS        DESCRIPTION
  D        OFF          DIAGNOSIS CONTROL BY CAPS SECTION (.DC)
  A        OFF          AUTOMATIC FAULT LOCATION
  G        OFF          GUIDED PROBE
  M        OFF          MANUAL TESTING


IC=         0           MAX   OF INDICTED IC'S TO USE INTERMEDIATE DRM
                        (ELSE PROBE IMMEDIATELY)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 6-4. DIAGNOSTIC Mode DIAGNOSTIC MODES Page.

DIAGNOSIS CONTROL BY CAPS SECTION (D)

If this option is ON then the diagnosis sub-mode selected for each CAPS section depends on the configuration established previously in the .DC file as described in para 6.2.6.

AUTOMATIC FAULT LOCATION (A)

If this option is ON, then AFL will be activated when the DIAGNOSE mode is first run or whenever the START button or ABORT button is depressed. In addition, immediate transfers to AFL obtained by the AUTOMATIC Button will be allowed only if this option is ON.

GUIDED PROBE (G)

If this option is ON, then the use of the Guided Probe/Finds is enabled. If it is ON and the A option (above) is OFF, then the Guided Probe will be activated when the DIAGNOSE mode is first run or whenever the START button is depressed. In addition, transfers to the Guided Probe via the GUIDED PROBE button or through the AFL Probe link are allowed only *if* this option *is* ON. The Quick Lookup feature can be run only if this option is ON and the A option is OFF.

MANUAL TESTING (M)

If this option is ON and both the A and G options are OFF, then Manual testing will be initiated when the DIAGNOSE mode is first run. In addition, transfers to the Manual sub-mode via the MANUAL button are allowed only *if* this option is ON.

IC=

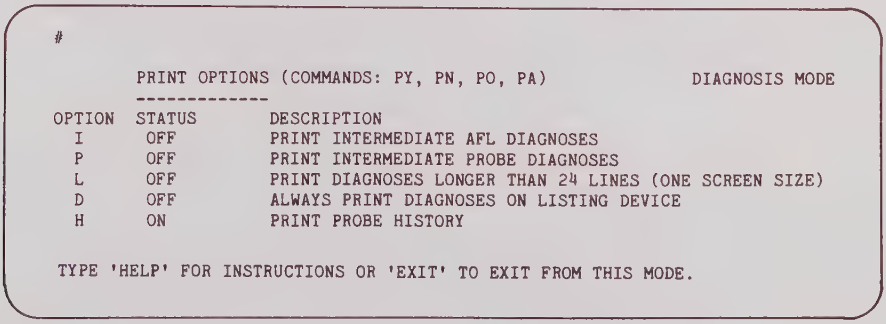
This option used only with Intermediate DRM. The IC= option is set to a value which is greater than the number of ICs the user wishes to clip. This allows the operator to set a limit on the time spent clipping IC's versus a direct transfer into Guided Probe/FINDS operation.

6.3.4 PRINT OPTIONS Page

The PRINT OPTIONS Page is the third page displayed (see Figure 6-5). The options on this page control the output of diagnostic messages in the following manner:

Print Intermediate AFL Diagnosis (I)

If this option is ON, AFL intermediate diagnosis messages will be output to the TEC display. Intermediate AFL diagnosis messages may occur after an AFL initial lookup or after an AFL Final Simulation run if the probe link is being used.



```
#

        PRINT OPTIONS (COMMANDS: PY, PN, PO, PA)          DIAGNOSIS MODE
        -------------
OPTION  STATUS      DESCRIPTION
  I      OFF        PRINT INTERMEDIATE AFL DIAGNOSES
  P      OFF        PRINT INTERMEDIATE PROBE DIAGNOSES
  L      OFF        PRINT DIAGNOSES LONGER THAN 24 LINES (ONE SCREEN SIZE)
  D      OFF        ALWAYS PRINT DIAGNOSES ON LISTING DEVICE
  H      ON         PRINT PROBE HISTORY

 TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 6-5. DIAGNOSE Mode PRINT OPTIONS Page.

Print Intermediate Probe Diagnosis (P)

If this option is ON, Probe intermediate diagnosis messages will be output to the TEC display. These are probe final diagnostic messages that are normally output just prior to entering FINDS, but may be suppressed by turning the P option OFF.

Print Diagnoses Longer Than 24 Lines (L)

In both intermediate and final diagnoses, the number of lines of data may potentially be very long - filling several TEC screens. The L option, if OFF, limits the part of any AFL or Guided Probe diagnosis seen by the operator to one TEC screen. If ON, the entire diagnosis is displayed on the TEC a page at a time.

Always Print Diagnosis on Listing Device (D)

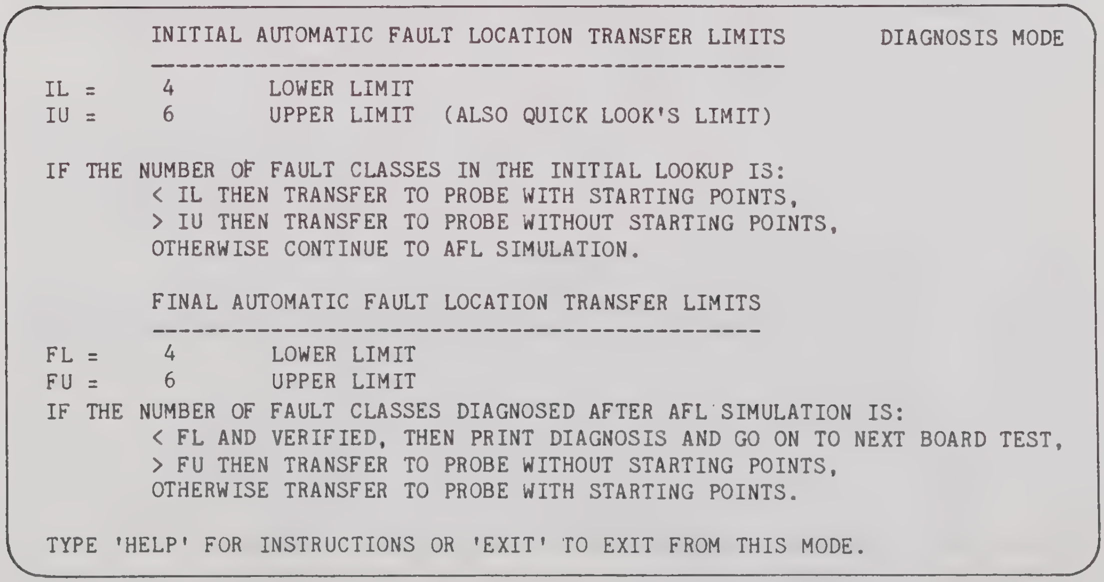
This option, if ON, causes any CAPS/APG diagnosis that appears on the TEC screen (in any diagnostic mode) to be duplicated on the listing device (LI on the FILE USAGE Page).

Print Probe History (H)

During the Guided Probe operation, a history of the points probed is saved and may be printed on the listing device if the H option is ON. If FINDS is OFF, the printout occurs immediately after the probe diagnosis (if the XMIT button is depressed). If FINDS is ON, the printout occurs immediately after the FINDS diagnosis by depressing the XMIT button.

6.3.5 AFL TRANSFER LIMITS Page

The AFL TRANSFER LIMITS Page is the fourth page displayed (see Figure 6-6). The four numeric parameters on this page control possible transfers from AFL to the guided probe/FINDS. The first two are invoked



```
        INITIAL AUTOMATIC FAULT LOCATION TRANSFER LIMITS      DIAGNOSIS MODE
        ------------------------------------------------
IL =     4       LOWER LIMIT
IU =     6       UPPER LIMIT  (ALSO QUICK LOOK'S LIMIT)

IF THE NUMBER OF FAULT CLASSES IN THE INITIAL LOOKUP IS:
        < IL THEN TRANSFER TO PROBE WITH STARTING POINTS,
        > IU THEN TRANSFER TO PROBE WITHOUT STARTING POINTS,
        OTHERWISE CONTINUE TO AFL SIMULATION.

        FINAL AUTOMATIC FAULT LOCATION TRANSFER LIMITS
        ----------------------------------------------
FL =     4       LOWER LIMIT
FU =     6       UPPER LIMIT
IF THE NUMBER OF FAULT CLASSES DIAGNOSED AFTER AFL SIMULATION IS:
        < FL AND VERIFIED, THEN PRINT DIAGNOSIS AND GO ON TO NEXT BOARD TEST,
        > FU THEN TRANSFER TO PROBE WITHOUT STARTING POINTS,
        OTHERWISE TRANSFER TO PROBE WITH STARTING POINTS.

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 6-6. DIAGNOSE Mode AFL TRANSFER LIMITS Page.

after an initial AFL Lookup, while the last two are invoked after the final AFL simulation run. One option (IU=) also applies to the probe's Quick Lookup transfer limits. Each parameter operates as follows:

LOWER LIMIT (IL=)

This number represents the initial lower limit. If, after the AFL initial lookup, the number of possible fault classes is less than this number, then a transfer to the Guided Probe with starting points is made. If the number of fault classes is greater than or equal to this number, then a decision based on the IU quantity (described below) is made.

UPPER LIMIT (IU=)

This number represents the AFL initial upper limit. If the number of fault classes after AFL initial lookup is greater than IL and IU, then a transfer to the Guided Probe/FINDS without starting points is performed. If, on the other hand, the number of fault classes is greater than or equal to IL and less than or equal to IU, then AFL simulation is invoked.

When using the Quick Lookup feature (PROBE sub-mode enabled and AFL sub-mode disabled), the IU= option specifies the maximum number of starting points that can be passed through the probe link to the computer-guided probe. If the number specified is exceeded, no starting points are passed to the probe; instead, the guided probe is entered directly without starting points.

LOWER LIMIT (FL=)

This number represents the AFL final lower limit. If the number of remaining fault classes after AFL simulation is less than this number and they are verified, then a final diagnosis is presented and testing of the next device is initiated. If the number of fault classes is greater than or equal to FL, then a further transfer decision is made based on the FU parameter described below.

UPPER LIMIT (FU=)

This number represents the AFL final upper limit. If the number of fault classes is greater than or equal to FL and less than or equal to FU, then a transfer to the Guided Probe with starting points is taken. If, however, the number of fault classes is greater than FU then a transfer to the Guided Probe without starting points is made.

6.3.6 GUIDED PROBE OPTIONS Page

The GUIDED PROBE OPTIONS Page is the fifth page displayed (see Figure 6-7). Options on this page may be used only if the PROBE sub-mode is enabled. Each option operates as follows:

Use Starting Points File (S)

If this option is ON, then user specified starting points will be used by the guided probe. User starting points specify the points at which the guided probe can be forced to start probing under failure conditions specified by a user. User starting points (.US) files may be prepared by the EDIT Mode and translated by the P sub-mode of PREPARE to obtain the binary data base required for use in the DIAGNOSE mode (refer to Para. 5.2.7).

Use Jumping Points File (J)

If this option is ON, then user specified jumping points will be used by the guided probe. User jumping points specify the next points to be probed if a failure is detected at a user defined point during probing (branch on errors only). User jumping point (.UJ) files may be prepared by the EDIT Mode and translated by the P sub-mode of PREPARE to obtain the binary data base required for use in the DIAGNOSE Mode (refer to Para 5.2.7).

Use Pulse Discrimination File (D)

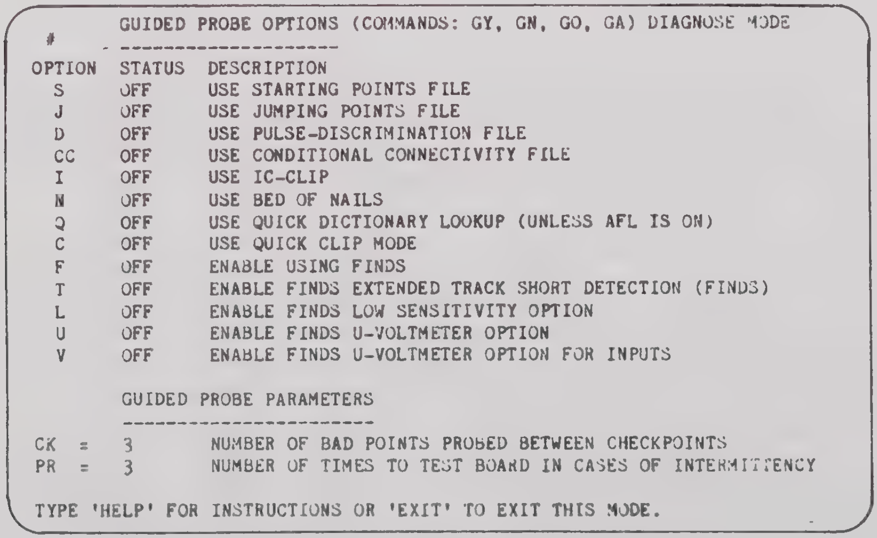
This option is for use with the GenRad pulse-catching probe only. If the option is ON, then the contents of a user specified pulse discrimination (.UD) file will be used by the guided probe. The file allows a user to resolve conflicts that can occur when a good UUT and the simulation data base mismatch because of inconsequential pulses detected by the sensitive GenRad pulse-catching logic probe.

If the option is OFF, then all pulse catching by the guided probe is ignored. Setting D=OFF is equivalent to using an IGNORE ALL .UD file; all pulse catching is ignored.

User-prepared pulse discrimination files may be created by the EDIT mode and translated by the P sub-mode of PREPARE to obtain the binary data base required for use in the DIAGNOSE Mode (refer to Para. 5.2.7). Pulse discrimination files may also be created automatically by use of the NODAL VERIFICATION mode (refer to Appendix D). NODAL VERIFICATION created .UD files are automatically translated; the PREPARE mode, P sub-mode is not used.

Use Conditional Connectivity (CC)

If this option is ON the .UC file (para 5.2.7) is used and the conditional connectivity feature is enabled. Conditional connectivity reduces the number of points that need to be probed when a node is driven by more than one set of inputs.



```
        GUIDED PROBE OPTIONS (COMMANDS: GY, GN, GO, GA) DIAGNOSE MODE
 #      ---------------------
OPTION  STATUS  DESCRIPTION
  S     OFF     USE STARTING POINTS FILE
  J     OFF     USE JUMPING POINTS FILE
  D     OFF     USE PULSE-DISCRIMINATION FILE
  CC    OFF     USE CONDITIONAL CONNECTIVITY FILE
  I     OFF     USE IC-CLIP
  N     OFF     USE BED OF NAILS
  Q     OFF     USE QUICK DICTIONARY LOOKUP (UNLESS AFL IS ON)
  C     OFF     USE QUICK CLIP MODE
  F     OFF     ENABLE USING FINDS
  T     OFF     ENABLE FINDS EXTENDED TRACK SHORT DETECTION (FINDS)
  L     OFF     ENABLE FINDS LOW SENSITIVITY OPTION
  U     OFF     ENABLE FINDS U-VOLTMETER OPTION
  V     OFF     ENABLE FINDS U-VOLTMETER OPTION FOR INPUTS

        GUIDED PROBE PARAMETERS
        -----------------------
CK  =   3       NUMBER OF BAD POINTS PROBED BETWEEN CHECKPOINTS
PR  =   3       NUMBER OF TIMES TO TEST BOARD IN CASES OF INTERMITTENCY

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 6-7. DIAGNOSE Mode GUIDED PROBE OPTIONS Page.

Use IC-Clip (I)

If this option is ON, one of the DRM I 16 or 24-pin IC clips may be used with the computer-guided probe. If the system is equipped with the DRM II and the I Option is ON, any of the clips, with the exception of the 40-pin clip, may be used. The IC clip connects to CLIP B of either DRM. The guided probe/clip asks users to clip rather than probe ICs. The clip enables a compilation of IC data which can be used as needed by the probe/clip. This speeds up the probing process by eliminating the necessity of repeated clipping.

NOTE

The IC clip always connects to CLIP B.

Use Bed-of-Nails (N)

If this option is ON, the Bed-of-Nails feature will be enabled. This feature eliminates manual probing of nodes that are connected to external Sensor pins (NAILS) during the normal probing sequence. It is ideally suited for circuit nodes that have a high degree of fan-out. When turned ON, all nodes that have attached Sensor pins and do not have the NONAIL attribute described in Para. 2.3.5.14 will be treated as NAILS.

Use Quick Dictionary Lookup (Q)

If this option is ON, and the AFL sub-mode is OFF, then a Quick Dictionary Lookup will be performed. Starting points resulting from the lookup may be passed to the guided probe through the probe link if the limit set by the IU= option is not exceeded. Starting points obtained in this manner, cause the probe to start probing at internal board failure points thus speeding-up the probing sequence. The data base files required to run the Quick Lookup feature are made when the PREPARE Mode Q sub-mode is run (refer to Para. 5.2.8).

Quick Clip (C)

If this option is ON, the IC-Clips may be used with the guided probe as in option I. However, Quick Clip looks at the first external failing test and only builds the data base up to this point. Since clipping will be compared with the simulation only for the test steps up to and including the failing test, not for the whole test program, the clipping speed is greatly increased. As a result of this limited clipping, an output designated as "good" during Quick Clip, is good only up to the failing point, not for the entire test program.

Using Quick-Clip on a UUT which is intermittent is not advised, because the failing test may vary in these cases, possibly resulting in an erroneous diagnosis.

Enable Using FINDS (F)

When this option is ON, the FINDS software is enabled. Because FINDS uses either DRM I or DRM II, it is necessary that the SETUP mode SYSTEM CONFIGURATION Page DR option is set to DR=1 or DR=2 before running FINDS. If the option does not agree with the hardware configuration, an error message is generated. FINDS allows the diagnosis of such errors as:

Shorts/opens between an IC pin and power or ground
Shorts/opens between adjacent IC pins
Faulty IC's
Shorts between tracks or an etch board.

If the F Option is OFF, FINDS cannot be used.

Enable FINDS Extended Track Short Detection (T)

If this option is ON, and FINDS is enabled, suspected track shorts, assumed by FINDS, can be analyzed further by allowing an operator to check for continuity between the original nodal point and other points on the node or any other random point on the board. If the T option is OFF, FINDS will diagnose the problem as a suspected track short, and not attempt to analyze the problem further.

Low Sensitivity Option (L)

When this option is ON, and FINDS is enabled, low power devices (such as CMOS) can be checked for proper current flow by increasing the path resistance and thus the voltage drop. This option should not be used when checking high power devices (such as TTL) as it decreases the chances of detecting bad IC's, and also increases the total number of voltage measurements required.

Enable FINDS Microvoltmeter Option (U)

If this option is ON, and FINDS is enabled, the system verifies that each IC output connected to the faulty node is functioning properly. This applies to either DRM I or DRM II.

Enable FINDS Microvoltmeter Option for Inputs (V)

If this option is ON, and the system is equipped with DRM II, the IC inputs can be checked with the DRM II microvoltmeter probe. The setting of this option has no effect on a system without DRM II.

The GUIDED PROBE OPTIONS Page also includes the following GUIDED PROBE PARAMETERS Options:

Number of Bad Points Probed Between Checkpoints (CK=)

While probing in the PROBE sub-mode, the guided probe occasionally checks the accuracy of the operator by asking that both ends (input and output) of a node be checked. This operation is called "checkpointing" and its frequency is controlled by the CK= option. CK=6 for example, then a checkpoint will be taken at every six bad points that are encountered while probing. If the Bed-of-Nails feature is ON, no checkpointing is accomplished regardless of the setting of the CK= option.

Number of Times to Test Intermittent Boards (PR=)

Devices under test can behave intermittently. Probing can be difficult under these circumstances since the guided probe depends on the device failing repeatedly and consistently. Whenever an intermittent condition is detected, the test program is run PR times for each point probed. The earliest failing test is then used by the guided probe.

6.3.7 LOGGING OPTIONS Page

The LOGGING OPTIONS Page is the sixth page displayed (see Figure 6-8). This page allows the results of CAPS/APG runs to be saved in a log file as the DIAGNOSE Mode is run. Methods of initializing and collecting data in the log file and running the LOG Mode are described in Appendix C. The DIAGNOSE Mode Log Options operate as follows:

Logging Enable (L)

If the L option is ON, and a current logging file exists for the board, the logging function will be enabled as the DIAGNOSE Mode is run. Diagnostic data from a CAPS/APG section can then be saved in the log file. Specific types of data such as the Date, Diagnosis, First Failing Test, Comments, Probe History, Fault Classes Detected, etc. may be saved in the file. A complete description of the information that can be saved and how to do it is contained in Appendix C.

Operator Comment Enable (C)

If this option is ON (together with L), then operator inserted comments (such as the board serial number of lot number) may be inserted as the DIAGNOSE Mode is run. A one line comment may be entered through the TEC Terminal or by Teletype in response to a prompt message at the appropriate time. Details are contained in Appendix C. If the C option is OFF, operator comments cannot be entered into the log file.

Device on Which Log File is Located (LD=)

This is the disk device that is to contain the Log (.DL) file. If a device is not specified, DSK: is assumed.

6.3.8 SELECT MISCELLANEOUS OPTIONS Page

The SELECT MISCELLANEOUS OPTIONS Page is the seventh page displayed (see Figure 6-9). The following options are available on this page:

Allow Full Monitor Using From Control Panel (M)

This and the next option control the use of the KEYBOARD CONTROL button and the commands that can be given from the operator's page during testing. If neither the M nor the C option is ON, then depressing the KEYBOARD CONTROL button will have no effect during testing. If the M option is ON, then when the KEYBOARD CONTROL button is depressed, the operator may change any of the parameters on the OPERATOR CONTROLS page (see Figure 6-10) or use the 'EXIT' command. This will allow the operator to use any of the modes within the CAPS/APG system.

Allow Limited Monitor Usage From Control Panel (C)

If the M option is OFF and the C option is ON, then when the KEYBOARD CONTROL button is depressed, the operator may change any of the parameters on the OPERATOR CONTROLS page. He may not, however, use the 'EXIT' command. He is restricted to remain in the DIAGNOSE mode of operation and may depress the START button or use the RUN command to resume testing.

Allow PAUSE Statements (P).

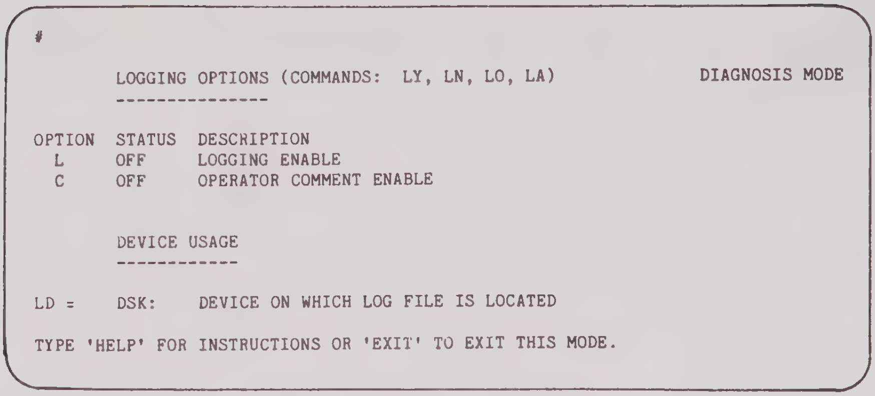
If this option is ON, then PAUSE statements within the test program will be executed; otherwise, they are treated as NOP (no-operation) instructions.

Allow WRITE Statements (W).

If this option is ON, then WRITE statements within the test program are executed; otherwise, they are treated as NOP instructions.

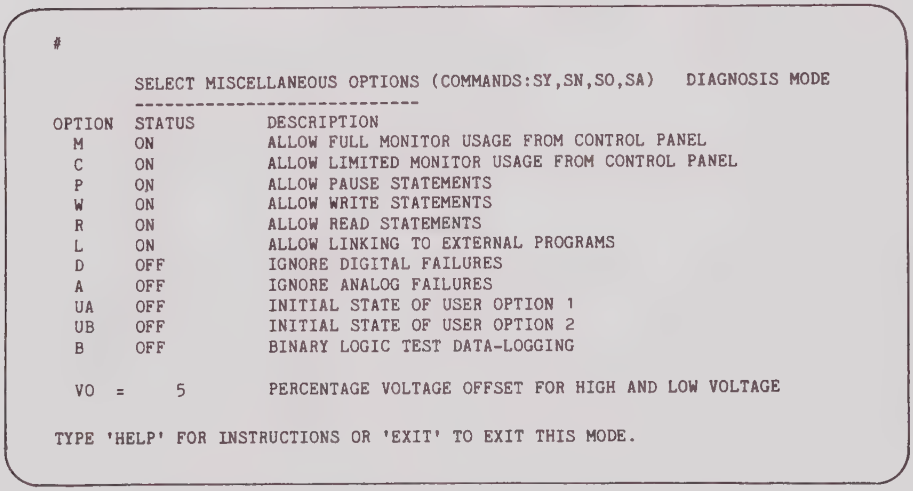
Allow READ Statements (R).

If this option is ON, then READ statements within the test program are executed; otherwise, they are treated as NOP instructions.



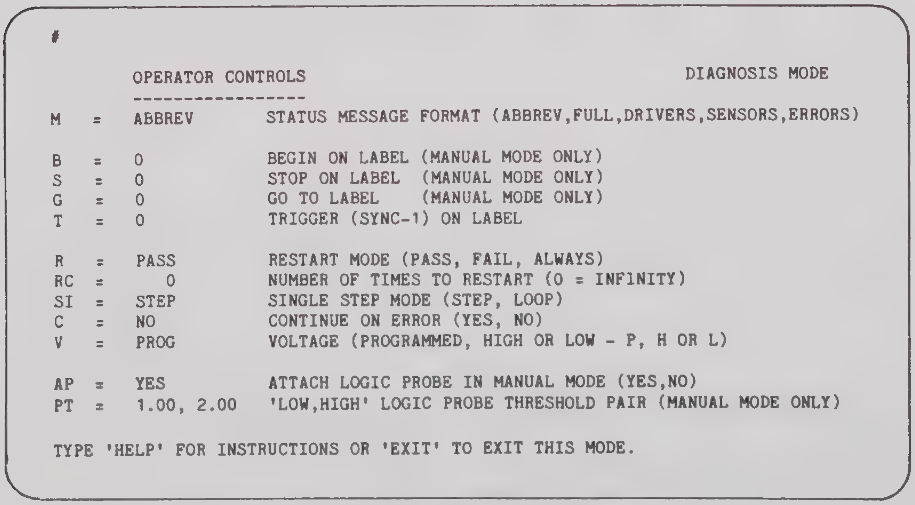
```
#

        LOGGING OPTIONS (COMMANDS:  LY, LN, LO, LA)           DIAGNOSIS MODE
        ---------------

OPTION  STATUS  DESCRIPTION
  L     OFF     LOGGING ENABLE
  C     OFF     OPERATOR COMMENT ENABLE


        DEVICE USAGE
        ------------

LD =    DSK:    DEVICE ON WHICH LOG FILE IS LOCATED

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 6-8. DIAGNOSE Mode LOGGING OPTIONS Page.



```
#

        SELECT MISCELLANEOUS OPTIONS (COMMANDS:SY,SN,SO,SA)   DIAGNOSIS MODE
        ----------------------------
OPTION  STATUS       DESCRIPTION
  M     ON           ALLOW FULL MONITOR USAGE FROM CONTROL PANEL
  C     ON           ALLOW LIMITED MONITOR USAGE FROM CONTROL PANEL
  P     ON           ALLOW PAUSE STATEMENTS
  W     ON           ALLOW WRITE STATEMENTS
  R     ON           ALLOW READ STATEMENTS
  L     ON           ALLOW LINKING TO EXTERNAL PROGRAMS
  D     OFF          IGNORE DIGITAL FAILURES
  A     OFF          IGNORE ANALOG FAILURES
  UA    OFF          INITIAL STATE OF USER OPTION 1
  UB    OFF          INITIAL STATE OF USER OPTION 2
  B     OFF          BINARY LOGIC TEST DATA-LOGGING

  VO  =     5        PERCENTAGE VOLTAGE OFFSET FOR HIGH AND LOW VOLTAGE

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 6-9. DIAGNOSE Mode SELECT MISCELLANEOUS OPTIONS Page.



```
#

        OPERATOR CONTROLS                                     DIAGNOSIS MODE
        -----------------
M   =   ABBREV       STATUS MESSAGE FORMAT (ABBREV,FULL,DRIVERS,SENSORS,ERRORS)

B   =   0            BEGIN ON LABEL (MANUAL MODE ONLY)
S   =   0            STOP ON LABEL  (MANUAL MODE ONLY)
G   =   0            GO TO LABEL    (MANUAL MODE ONLY)
T   =   0            TRIGGER (SYNC-1) ON LABEL

R   =   PASS         RESTART MODE (PASS, FAIL, ALWAYS)
RC  =      0         NUMBER OF TIMES TO RESTART (0 = INFINITY)
SI  =   STEP         SINGLE STEP MODE (STEP, LOOP)
C   =   NO           CONTINUE ON ERROR (YES, NO)
V   =   PROG         VOLTAGE (PROGRAMMED, HIGH OR LOW - P, H OR L)

AP  =   YES          ATTACH LOGIC PROBE IN MANUAL MODE (YES,NO)
PT  =   1.00, 2.00   'LOW,HIGH' LOGIC PROBE THRESHOLD PAIR (MANUAL MODE ONLY)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure 6-10. DIAGNOSE Mode OPERATOR CONTROLS Page.

Allow Linking to Ext. Programs (L).

If this option is ON, then BRANCH statements in the test program that invoke external programs are executed; otherwise the link request is treated as a NOP instructions.

Ignore Digital Failures (D).

If this option is ON, then failures on digital drivers and sensors are ignored; otherwise, a normal failure behavior pattern will be followed.

Ignore Analog Failures (A).

If this option is ON, then failures on any analog equipment will be ignored; otherwise, a normal failure behavior pattern will be followed.

Initial User Option 1 (or 2) State (UA= or UB=).

The UA= and UB= options may be set to either 1 or 0. Both options establish the initial states of software registers. The UA= option sets the initial state of the USOPT1 register, while the UB= option sets the initial state of the USOPT2 register. Refer to the Programming Instructions (1792-0576) for a complete description of register usage.

Binary Logic Test Data-Logging (B)

This option may be used only in the MANUAL sub-mode. If the option is ON, then data logging in a user file of failing test vectors that results from a UUT failure can be accomplished. Operation in this mode requires the preparation of a special user program prepared in DEC PAL8 Assembly Language to change the current bit vector file format. This option should therefore be used only be experienced PAL8 Assembly Language Programmers. Detailed procedures are available in a special Applications Note that may be obtained through GenRad Software Services.

Percentage Voltage Offest (VO=)

This option specifies the percentage from normal by which to modify the test program SET PS Voltage specification if the V= option on the operator controls page has been set to HIGH (+) or LOW (-). This option has no effect if the operator controls page V= option has been set to PROGRAMMED.

6.3.9 OPERATOR CONTROLS Page

The OPERATOR CONTROLS page (Figure 6-10) is seen from two perspectives: as the last page in the DIAGNOSE mode of operation and as the only page of a mode obtained by depressing the KEYBOARD CONTROL button. The values on this page may be initially set through DIAG, then modified through KEYBOARD CONTROL. The options on this page apply mostly to the MANUAL sub-mode of operation. Only R (Restart Mode), RC (restart Count), T (Trigger), and V (Voltage) are used by the PROBE and AFL sub-modes. The following commands may be issued form this page:

Status Message Format (M).

When a failure occurs (or when single stepping in the MANUAL sub-mode), a status message is displayed on the TEC screen. This initial status message may be formatted in one of five ways as dictated by the M option:

1. ABBREV - an abbreviated display of the states of all drivers and sensors.

2. FULL - a full display of all drivers and sensors including their state as a driver or sensor, present value, and associated pname.

3. DRIVERS - same as FULL, but for drivers only.

4. SENSORS - same as FULL, but for sensors only.

5. ERRORS - same as FULL, but for pins in error only.

Begin on Label (B).

If this number is non-zero, then all MANUAL sub-mode testing will commence with the indicated labelled statement in the test program. If B is set to 0, then the normal testing sequence will be followed.

Stop on Label (S).

If this number is non-zero, then all MANUAL sub-mode testing will stop before executing the indicated labelled statement in the test program. If S is set to 0, then the normal testing sequence will be followed.

Go to Label (G).

If this number is non-zero, then all MANUAL sub-mode testing will start at the first test statement (if B=0) or the statement indicated by B, and ignore all failures until the label indicated by G is reached; that is, normal input stimuli will be applied until the statement label "G", but all failures will be ignored up to that point. Once "G" is reached, normal testing will be resumed.

Trigger on Label (T).

If this number is non-zero, a positive 1μs pulse will be emitted from the SYNC-1 terminal approximately 5μs before the beginning of execution of the labelled statement.

Restart Mode (R).

This option works in conjunction with the RESTART MODE control panel button. It has two choices in the PROBE or AFL sub-modes (PASS or ALWAYS) and three choices in the MANUAL sub-mode (PASS, FAIL, or ALWAYS). When the button is depressed in the PROBE or AFL sub-mode, the tester will perform a restart if the UUT passes and R=PASS or ALWAYS, and will go into PROBE or AFL if the UUT fails. In the MANUAL sub-mode, the tester will perform a restart on PASS, FAIL, or ALWAYS, respectively, depending on the state of the R option.

Restart Count (RC).

This number dictates the number of restarts (passes) to be performed under the restart mode. If RC=0, the restart count is considered infinite.

Single Step Mode (SI).

This option works in conjunction with the STEP MODE control panel button and has two choices; STEP or LOOP. When the button is depressed in the MANUAL sub-mode, the tester will operate in single STEP or single LOOP mode according to the setting of the SI option .

Continue on Error (C).

If this option is OFF, then the instant that any non-ignored digital or analog failure occurs all test equipment and power supplies are disconnected and shut-down. This is for the purpose of protecting the UUT. If this option is ON, then test equipment remains connected and power supplies remain on when a failure occurs. This, as indicated by the CONTINUE light being on, allows the operator to continue from the point of failure when working in the MANUAL sub-mode.

Voltage (V).

If this option is set to PROGRAMMED (V= PROG), then the voltage levels specified in all test program SET PS statements will be the levels that are compared with the SETUP Mode Power supply configuration values in order to select the correct user power supply output levels for each power supply. If this option is set to HIGH (or LOW), then all SET PS statement levels will be offset HIGH (+) or LOW (-) by the numeric percentage specified by the VO= option on the Select Miscellaneous Options page before being compared with the SETUP Mode power supply configuration values.

Attach Logic Probe (AP=).

The AP= and PT= options apply only if the GenRad Pulse-Catching logic probe is being used. The SETUP Mode SYSTEM CONFIGURATION Page LP= Option must also be set to LP=YES for this option to work. The probe can be operated in the MANUAL sub-mode to display transitions across threshold boundaries together with the final state encountered in a test step. If the test system is being operated in the Single STEP Mode, then values for each test step can be displayed; otherwise, the values of the last test step will be displayed when a failure occurs. The AP= option must be set to YES in order to enable this feature.

Logic Probe Threshold Pair (PT=).

This option applies only to the GenRad pulse-catching probe. If the PA= option has been enabled (PA=YES), then the PT= option will specify the low, and high threshold values to be assigned to the probe in the MANUAL sub-mode. Values must be specified in the form:

PT=0.7, 2.0

Where the low threshold value is always listed first.

6.4 DIAGNOSTIC ERROR MESSAGE FORMATS

6.4.1 General

Diagnostic error message formats will differ depending on whether the message is a result of a fault diagnosis from either the Probe/FINDS and AFL, or whether the MANUAL sub-mode is being used by itself. Some sections of the messages generated in the AFL and PROBE/FINDS sub-modes may also differ, depending on whether the X-1 passive probe, GenRad pulse-catching probe, or IC clip are being utilized as the computer-guided (smart) probe. Diagnostic messages are described in the following paragraphs.

6.4.2 AFL/PROBE/FINDS Messages

The general format used in AFL/PROBE/FINDS messages is shown in Figure 6-11. These diagnostic messages are always displayed on the TEC Terminal. Hard copy printouts may also occur if the DIAGNOSE Mode PRINT OPTIONS D Option is ON. The general diagnostic message formats are as follows:

BOARD and TEST PROG Name

Both entries currently reflect the name assigned by the DIAGNOSE Mode FILE USAGE Page TP= option.

DATE:

The DATE listed is the date on which the diagnosis has been performed (the date listed by the CCL .DA command prior to entering the monitor top level).

DIAGNOSIS FROM:

This entry reflects the method used to obtain the diagnosis. The Diagnosis may be from one of the following diagnostic techniques:

GUIDED PROBE
FINDS
AUTOMATIC (AFL) - Message

If the diagnosis is from AFL, one of the following additional messages will also occur:

1. AUTOMATIC-INITIAL - an AFL initial lookup has been performed and the diagnosis represents the possible faults from the lookup.

2. AUTOMATIC-VERIFIED - an AFL simulation has been performed and the diagnosis indicates the failure(s) whose fault simulation exactly match the faulty board's behavior.

3. AUTOMATIC-VERIFIED, OTHER FAULTS PRESENT - an AFL simulation has been performed and the diagnosis indicates the faults whose failures are a subset of the board's failures. This means that the failure(s) in the diagnosis are verified as being present and in addition that there are other failures on the board.

4. AUTOMATIC-PROBABLE - an AFL simulation has been performed and the diagnosis indicates that the failures in the diagnosis are the probable cause of the board's failure. This is due to the fact that the simulated faults did not exactly match the boards behavior but were "close enough" to justify indicating them as probable causes.

```
BOARD:    SAMPLE             TEST PROG:  SAMPLE        DATE:  3/11/77
DIAGNOSIS FROM:  GUIDED PROBE                          FAILING TEST  62
COMMENT:  NODE ERROR
                                                CONNECTIONS
IC2.2*    (7092)   NODE MIDRANGE BUT ACTIVE
                   NODE MISSING PULSE
                   NODE EXHIBITS HALF PULSE     IC7.2(936)    IC7.4(936)
                                                IC7.8(936)    IC7.6(936)
                                                IC7.5(936)    IC1.12(946)
                                                IC1.4(946)

                         OR

COMMENT:  BROKEN CONNECTION

IC2.2*    (7072)   BROKEN CONNECTION WITH    IC4.3(936)
                       OTHER CONNECTIONS        IC8.3(7474)   IC6.7(936)
                                                IC10.11(7474) IC10.8(7474)
                                                IC6.8(936)    IC7.2(936)
IC4.3*    (936)    PIN STUCK MIDRANGE            (SEE ABOVE)
```

PROBE MESSAGES

```
BOARD:    SAMPLE           TEST PROG:   SAMPLE       DATE: 2/8/79
DIAGNOSIS FROM:  FINDS                               FAILING TEST # 1
COMMENT:  PIN FAILURE/BAD IC

FINDS UNABLE TO IMPROVE DIAGNOSIS. OPERATIONS PERFORMED: -
    ADJACENT TRACK SHORTS

ORIGINAL PROBE DIAGNOSIS: -                   CONNECTIONS
                                              IC2.2  (7410)
IC3.16* (74LS240) NODE LOW BUT ACTIVE         IC3.8  (74LS240)


BOARD:    SAMPLE           TEST PROG:   SAMPLE       DATE:  2/8/79
DIAGNOSIS FROM:  FINDS                               FAILING TEST # 1
COMMENT:  TWO NODES SHORTED
                                              CONNECTIONS
IC3.16* (74LS240) NODE                        IC2.2  (7410)
                  SHORTED TO                  IC3.8  (74LS240)
 .7               NODE                        GND
```

FINDS MESSAGES

```
BOARD:    SAMPLE             TEST PROG:   SAMPLE         DATE:   3/11/77
DIAGNOSIS FROM:  AUTOMATIC - PROBABLE                    FAILING TEST   1

                                                 CONNECTIONS
25*  (D/S 25)   D/S  STUCK LOW                   IC10.14(74193)  IC9.14(74193)
                                                 IC8.14(74193)   IC7.14(74193)
IC10  (74193)    SHORT BETWEEN PINS 10 AND 11
```

AFL MESSAGE

Figure 6-11. Typical PROBE/FINDS/AFL Diagnostic Message Formats.

FAILING TEST

When the diagnosis comes from the Guided PROBE/FINDS, this number is the test number for the diagnosed failure; that is, the failure(s) indicated in the diagnosis are true at the failing test # given and beyond. It is always the lowest test number at which an incorrect value was observed.

When the diagnosis comes from AFL initial or AFL final, the failing test # is the test number at which the UUT failed as viewed from external pins only.

COMMENT

The comment field appears in guided PROBE/FINDS messages only to indicate the general cause of a failure.

Some possible PROBE comments might include:

1. PROBE TERMINAL POINT - a bad connection to a device with no inputs such as a VCC, GND or PULLUP was found, or an incorrect value at an unconnected input pin was found.

2. BROKEN CONNECTION - a broken connection between an output and input pin along the same node was found.

3. BAD FEEDBACK LOOP - a failure somewhere along a logical feedback path was found; the diagnosis lists all possible failures.

4. DRIVER ERROR - A Driver pin was found to be faulty (most likely due to a short on the board.

5. NODE FAILURE - a circuit node was found faulty.

6. WIRED-AND NODE FAILURE - a circuit node failure was found at a node where several outputs are wired together in a WIRED-AND configuration.

7. ANALOG FAILURE-NO DIGITAL CAUSE - a non-digital failure occurred while running the test program, all possible digital causes were investigated (as dictated by the user's starting points file) and no digital causes were found.

8. ANALOG FAILURE-CAN'T CONTINUE - a non-digital failure occurred, but the system cannot continue because, either:

   a. The 'user starting points file' S option was OFF.

   b. No entry for the failure was found in the starting points file.

In addition to the above comments, one of the following three messages may also appear below the comment line:

1. EXTERNAL INTERMITTENT - the UUT's external outputs behaved intermittently during probing.

2. INTERNAL INTERMITTENT - nodes internal to the UUT behaved intermittently during probing.

3. INTERNAL AND EXTERNAL INTERMITTENT - a combination of (1) and (2) above.

Some possible FINDS comments might include:

1. TWO NODES SHORTED - Two nodes have been found shorted together.

2. SUSPECTED TRACK SHORT - A suspected track short has been detected. This message may occur if a suspected track short has

been detected and the FINDS Extended Track Short Detection (T) Option is OFF. FINDS does not analyze suspected track shorts further unless the T Option is ON.

3. PIN FAILURE/BAD IC - The IC pin has failed the voltage measurement test and should be considered bad.

4. FINDS DETECTED CONTINUITY -INCONSISTENT NODAL MEASUREMENT- Different test results have been obtained by the guided probe and FINDS. The guided probe detected different values at either end of the node. The clip has probably pulled up the node.

5. SHORTED TO POWER PIN - A node that is shorted to a power pin has been detected.

6. NO SHORT DETECTED - BAD IC? - The track short strategy failed to detect a short. Since the μ voltage test is not 100% conclusive, the system is suggesting that an IC may be bad.

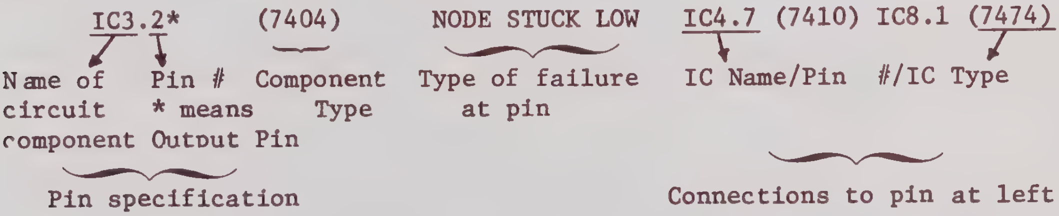
OPERATIONS PERFORMED

The Operations Performed portion of the FINDS message is displayed only when FINDS is unable to improve on the original probe diagnosis. The operations performed are:

ADJACENT TRACK SHORTS
μ-VOLTAGE
TRACK SHORTS
NONE

Diagnostic Messages

The remainder of the diagnosis printout contains a description of the failures being diagnosed. All lines have the same three field format as shown in the following example:



```
IC3.2*     (7404)     NODE STUCK LOW     IC4.7 (7410) IC8.1 (7474)
```

The first field is the pin at which the diagnosis is made and contains the ICNAME.# (TYPE) where ICNAME is the name of the IC for the faulty pin, the "." should be read as "pin",# is the pin number on the IC and TYPE is the type of component that the IC is. The last field is a list of pins connected to the pin in the first field. The format of these pins is the same as that in the first field.

In either the first or last fields, an external pin may be mentioned. If so, the format of the pin specification is:

EXTXYZ (D/S #)

where EXTXYZ is the name of the external and # is the pin system driver/sensor number.

The center column represents the actual failure present. Some possible printouts are:

1. NODE MIDRANGE BUT ACTIVE
   NODE MISSING PULSE
   NODE EXHIBITS HALF PULSE
   NODE EXHIBITS ILLEGAL PULSE
   NODE STUCK MIDRANGE

All of these messages apply only if the GenRad pulse-catching probe is being used. Additional messages of the same general type may also occur as the need dictates.

2. BROKEN CONNECTION WITH IC4.3(936)
   OTHER CONNECTIONS
   (see Figure 6-11)
   PIN STUCK MIDRANGE

The full printout is shown in Figure 6-11. Note that IC2.2 in that illustration is an IC output pin (*) that should be connected to input pin 3 of IC4, but is not because of a break. IC output pins are always identified by an * (usually on the left side of the message) while IC inputs are listed on the right side. The OTHER CONNECTIONS listing identifies other IC pins that are connected to the same broken node. Note, however, that there is no way of determining whether these connections are actually attached to the output (IC2.2) or input (IC4.3) side of the break. The last entry lists the type of error (PIN STUCK MIDRANGE) detected for the pin listed above (IC4.3).

3. NODE STUCK LOW
   NODE STUCK HIGH
   NODE LOW BUT ACTIVE
   NODE HIGH BUT ACTIVE

These four messages indicate that there is a problem associated with a node as a whole; that is, with all outputs and inputs at a circuit node. The actual failure, then, can be at the output pin in the first column, at any of the connection field pins, or along any of the etches connecting the points.

The STUCK LOW or HIGH indicates that the node is permanently stuck at logic "0" or "1", respectively. This can be due to a faulty IC (any one mentioned) or a short to a VCC or GND track. At this point, FINDS may be run to further resolve the problem, if desired.

The LOW (or HIGH) BUT ACTIVE indicates that the node is incorrectly low or high at the failing test # but that it is also capable of changing to the opposite state somewhere in the test program. This can be due to a faulty IC (any of the ones mentioned) or a short to some other signal path.

4. DRIVER STUCK LOW
   DRIVER STUCK HIGH
   DRIVER LOW BUT ACTIVE
   DRIVER HIGH BUT ACTIVE

These messages indicate the same failures as in (3), except for a Driver/Sensor pin.

5. INPUT STUCK LOW
INPUT STUCK HIGH

These two diagnoses indicate that there is a problem associated with an input pin as opposed to an output pin or node. STUCK LOW indicates a faulty IC or pin shorted to ground. STUCK HIGH indicates a broken connection to the driving output (most likely for TTL type logic), a faulty IC (less likely) or a short to VCC (least likely).

The test programmer may also choose to associate a text message with any pin. If the pin specification on the left of the printout has a diagnostic message associated with it, or if a message is associated with the NODE, then this message will appear under the third field for a diagnosis. For example, the text "A PULSE GENERATOR FEEDS IC8.9", may be included. Such messages are created in the .SR file using the % processing message option (refer to para. 2.3.5 and 2.3.5.6). The first "↑" in the text of a message signifies the beginning of a diagnosis message and each succeeding "↑" is replaced by a carriage return by the the Diagnosis Printer.

6.4.3 MANUAL Sub-Mode Messages

When operating in the MANUAL sub-mode, the system has the ability to output failure states in a number of ways. Figure 6-12 lists some examples of such displays. The figure represents the five ways in which the states of all logic driver/sensors may be displayed. The mode in which the system displays its error message can be controlled in two ways: through the M= option on the OPERATOR CONTROLS page of DIAG mode, and through the TEC keyboard (or Teletype) when single stepping or when a failure occurs. The MANUAL sub-mode diagnostic error message formats are as follows:

Digital Driver/Sensor Display

The five driver/sensor displays are named FULL, DRIVERS, SENSORS, ERRORS, and ABBREV.

All of these messages have similar headings: The top line displays the current test step number in the form:

TEST STEP X + Y OF NAME

where X is the last test statement label encountered in the test program, Y is the number of test steps since label X, and NAME is the name of the test program being run. If Y is 0, the "+Y" part will be omitted. To the right on the top line of the display is a cumulative total of the number of tests that have failed up to this point. On the second line, below the words "TEST STEP", the words "DRIVERS", "SENSORS" or "ERRORS" will appear when these types of displays are active (see Figure 6-12). Following this on the second line is a message of the form:

X PINS IN ERROR

where X is the number of driver/sensors with incorrect values. To the right on this line the word IGNORE indicates that one or more pins are ignored at the test.

```
TEST STEP                     10+     2 OF TJ1             2 TESTS FAILED
                           1 PINS IN ERROR                     IGNORE
CAPS       #          3

INH1       1 D1               INH2    2 D1       CLK      3 DO          4 DO
B          5 DO               C       6 DO       DOUT     7 S1<   FAKEIN 8 DO
          10 SO-
```

Full D/S Display

```
TEST STEP                     10+     2 OF TJ1            2 TESTS FAILED
           DRIVERS         1 PINS IN ERROR                     IGNORE
CAPS       #          3

INH1       1 D1               INH2    2 D1       CLK      3 DO          4 DO
B          5 DO               C       6 DO       FAKEIN   8 DO
```

Driver Display

```
TEST STEP                     10+     3 OF TJ1             2 TESTS FAILED
           SENSORS         1 PINS IN ERROR                     IGNORE
CAPS       #           4

DOUT       7 S1<                    10 SO-
```

Sensor Display

```
TEST STEP                     10+     3 OF TJ1             2 TESTS FAILED
           ERROR           1 PINS IN ERROR                     IGNORE
CAPS       #           4

DOUT       7 S1<
```

Error D/S Display

```
TEST STEP                     10+     2 OF TJ1             2 TESTS FAILED
                           1 PINS IN ERROR                     IGNORE
CAPS       #           3

DDDDD DSD*S   *****     *****     *****     *****    *****    *****    *****
11000 010 0
      ^   -
```

Abbreviated D/S Display

Figure 6-12. MANUAL Sub-Mode D/S Displays.

Below this line is a message of the form

```
CAPS #   X
```

where X is the test number which all CAPS printouts refer to. This message is present only at TEST LGC steps within CAPS recording sections.

The FULL display presents the states of all driver/sensors. In figure 6-12, for example the text:

```
INH1  1  D1
```

is the state of pin 1. "INH1" is the name of the pin as given in a previous ASSIGN PNAME statement; the "1" following it indicates D/S #1. The "D1" indicates that the pin is assigned as a Driver and is presently driving a "1" (or high level); the space following "D1" indicates that the pin's state is correct and not ignored.

Similarly, the entry:

```
DOUT  7  S1<
```

indicates that the pin named "DOUT" is D/S # 7, assigned as a sensor, and is reading a state of "1". The "<" indicates that this pin is in error; that is, it is reading a state other than the one expected. A "-" in this position, as in:

```
10 SO-
```

indicates that the pin is being ignored at the moment.

It should be noted that the number of D/S's assigned may be large enough so as to require two pages to be displayed. The second page (if it exists) can be viewed by depressing the XMIT key on the TEC keyboard. The existance of a second page is indicated by a ^^^^^ prompt on the bottom display line.

The DRIVERS display (Figure 6-12) is identical to the FULL display, but for pins assigned as drivers only. Similarly, the SENSORS display shows pins assigned as sensors in a FULL format. And finally, the ERRORS display lists all drivers and sensors that are in error in the same format as the FULL format.

The ABBREV format of the driver/sensor display enables sets of fifty D/S to be displayed three lines to a set. The first set corresponds to D/S 1-50, the second to D/S 51-100, etc. The first line of each set shows whether the pin is a driver (D), sensor (S) or unassigned (*). The second line shows the current sensed state of the pin and the third line depicts whether the pin is in error (^) or ignored (-).

Non-Digital Display

When a failure occurs at a non-digital test step or while single stepping through a non-digital test a different display is presented. Figure 6-13 is an example of a non-digital failure - in this case a DC voltage limit failure. The top line displays the failing test number and test program name. The second line displays the current variable being compared and its present value in exponential notation. The third line is displayed for a failure and depicts the low and high limits being compared against. Figures 6-14 and 6-15 depict step mode displays involving the arithmetic register (AR) and REG-10 respectively.

```
TEST STEP            20 of TJ1

DCV       IS -0. 132617E+000
LIMIT FAILURE -- LOW LIMIT -0.128750E+001      HIGH LIMIT   -0.121250E+001
```

Figure 6-13. Non-Digital Failure Message.

```
TEST  STEP                  0 +    1 OF PATMP

AR       IS          0.500000E+001
```

Figure 6-14. Step Mode Message For Arithmetic Register.

```
TEST STEP                   0 +    4 OF PATMP

REG-     10    IS          5.000
```

Figure 6-15. Step Mode Message For Register 10.

Probe Display

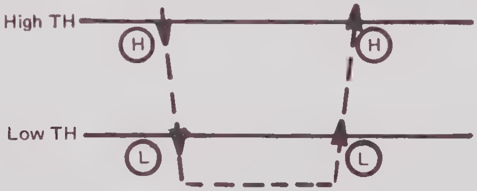
The GenRad pulse-catching probe can be used in the MANUAL sub-mode to display a value that represents the type and number of transitions that have occured across threshold boundaries together with the final level state encountered in a test step. If the test system is being operated in the Single Step Mode, then values for each test step can be displayed; otherwise, the values of the last test step will be listed when a failure occurs. This feature can only be used if the DIAGNOSE Mode Operator Controls page AP= option has been enabled, and logic probe low and high threshold values have been specified through use of the PT= option.

The MANUAL sub-mode probe message has the form:

PRB=HLLH H

Where:

HLLH Specifies the type of transition that has occurred. Transitions across threshold boundaries are coded as follows:



Where PRB=HLLH indicates that four transitions have occurred across threshold boundaries (High, Low, Low, and High). Dashes (-) indicate that transitions did not occur. For example, the message

PRB=HL--

specifies that only High to Low transitions have occurred.

The last letter in the probe message (following a space) indicates the final or ending level state encountered. The following letter codes may occur:

Code	Final State
M	Ended in the Middle (between Thresholds)
L	Ended Low
H	Ended High
B	Both High and Low thresholds have been detected (probably caused by an oscillation or threshold overlapping).

DCV probe measurements are also possible during a PAUSE in the test program (or whenever a status message is displayed) by placing the probe on the point to be measured and then depressing the PROBE button. The message format is

PROBE DCV = XY

Where XY is the nominal voltage level at the probed point.

6.5 COMPUTER-GUIDED PROBE/CLIP TESTING

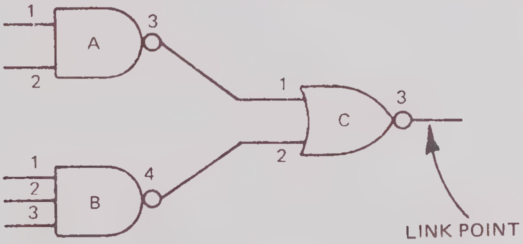
6.5.1 Guided Probe/Clip Testing Flow

The computer-guided probe/clip may be entered whenever the PROBE sub-mode of DIAGNOSE is enabled. If the Quick Dictionary Lookup feature is also enabled, the PROBE sub-mode will use two pieces of data obtained from UUT testing to select a starting probe point based on a fault dictionary lookup. This information includes the failing test number and the failing vector; that is, the test program step at which a failure first occurs and a list of the driver/sensors which failed at that test.

The Quick fault dictionary prepared from a CAPS/APG PREPARE mode Q sub-mode run is then used. This dictionary holds data showing how each modelled fault should be detected and is organized first by step number, then by failing vector. Thus, at testing time the entries in the fault dictionary for a UUT's failing test are extracted. Only those entries whose failing vectors match the UUT's failing vector are considered possible causes of the UUT's failure.

In other words, the Quick fault dictionary lookup pulls out those modelled faults which, in simulation, failed at the same step and in the same way as the failing UUT. This process serves to narrow down the number of faults possibly responsible for a particular UUT's failure from hundreds or thousands to about ten. In some cases the resolution at this point might be quite good - that is, the number of possible faults will be small due to the fact that different faults manifest themselves in different ways. In other cases, where the UUT fails early in the test program, or few outputs from the board are visable, the resolution may be so poor that it is not feasable to pass starting points through the probe link to the probe.

The Probe Link performs the task of instructing the Guided Probe as to where to probe initially, based on the fault classes under investigation. The Link does this by selecting one node from each fault class. A fault class is a group of one or more individual faults (stuck at zero, stuck at one, short to another pin) which are indistinguishable as observed from a UUT's external behavior. Although the Guided Probe can look at any one of the pins included in a fault class, it must be given a single point to look at, one which will be bad if any of the individual faults in the fault class is present. For example, consider the following circuit:



One fault class for this circuit is:

A.1-0 A.2-0 A.3-1 B.1-0 B.2-0 B.3-0 B.4-1
C.1-1 C.2-1 C.3-0

involving 10 individual faults and 8 different nodes. It can be seen, however, that C.3 is "furthest down-stream" of all the pins in the fault class. In fact, this is the very point that the Link will pass to the Guided Probe because it will be bad if any of the faults in the fault class are bad. Because of the way in which fault collapsing is done and the way in which the Link operates, the selection of the correct fault from a fault class can always be performed.

The Guided Probe/clip, when instructed by the Link, first looks at the points given it. If any of these points are bad, the Guided Probe follows its normal mode of isolating a failure - but starts at the bad point. This is a great advantage over the normal Guided Probe start-up procedure which involves tracing back from a failing external output. The use of the Link allows the Guided Probe to jump immediately to the area of failure and often saves many minutes of probing.

The Quick Lookup option only functions if the AUTOMATIC (AFL) sub-mode is not enabled. If AFL is ON, then AFL will be run, when a board failure occurs. Through proper use of the DIAGNOSE Mode AFL TRANSFER LIMITS Page IL=, IU=, FL=, and FU= options it is possible to transfer from AFL to Intermediate DRM or the Guided-PROBE/FINDS with probe starting points (refer to Para. 6.7) or to Guided-Probe/FINDS without starting points. In order to do this, the PROBE sub-mode must also be enabled

The IU= option on the AFL TRANSFER LIMITS Page also allows a user to select the maximum number of starting points to be passed to the probe through the probe link if the AFL sub-mode is OFF, the PROBE sub-mode is ON, and the Quick Lookup feature is enabled. If the quick initial lookup generates more than this number, then the guided probe will be entered without passing starting points through the link. Under these conditions, the guided probe will start probing back from failing external UUT output pins unless user starting points are supplied.

The computer-guided probe/clip overall flow is shown in Figure 6-16. The flow shown is initiated by depressing the control panel START Button. Board testing is performed in all sub-modes. If the PROBE sub-mode has been selected, and a board failure occurs, the probe sub-mode will be entered if the AFL sub-mode is OFF. If AFL is ON, then AFL will be entered and run. Starting points resulting from an initial or final AFL run may be passed through the probe link to Intermediate DRM or to the probe.

If AFL is OFF, the Quick Lookup feature of the PROBE sub-mode may be selected to quickly pass starting points to Intermediate DRM or to the probe. The operational flow once the guided probe/clip has been entered is as follows:

1. A probing loop is entered and a probe/clip starting point is determined. The first probe point may be a starting point passed through the probe link as the result of a quick lookup, an AFL lookup, a starting point obtained from a user's starting point file, or a point selected by the guided probe/clip itself.

2. A message then instructs the operator to place the probe/clip on the point to be probed/clipped and depress the control apnel PROBE Button (or optional footswitch).

3. If an IC cannot be clipped for any reason, the operator may depress the XMIT Button, and the guided probe/clip will request that the IC be probed. The software will also remember that the IC is not to be clipped again during this test sequence. When the START Button is again depressed to start another test sequence, the IC may be clipped again (unless the XMIT Button is again depressed).

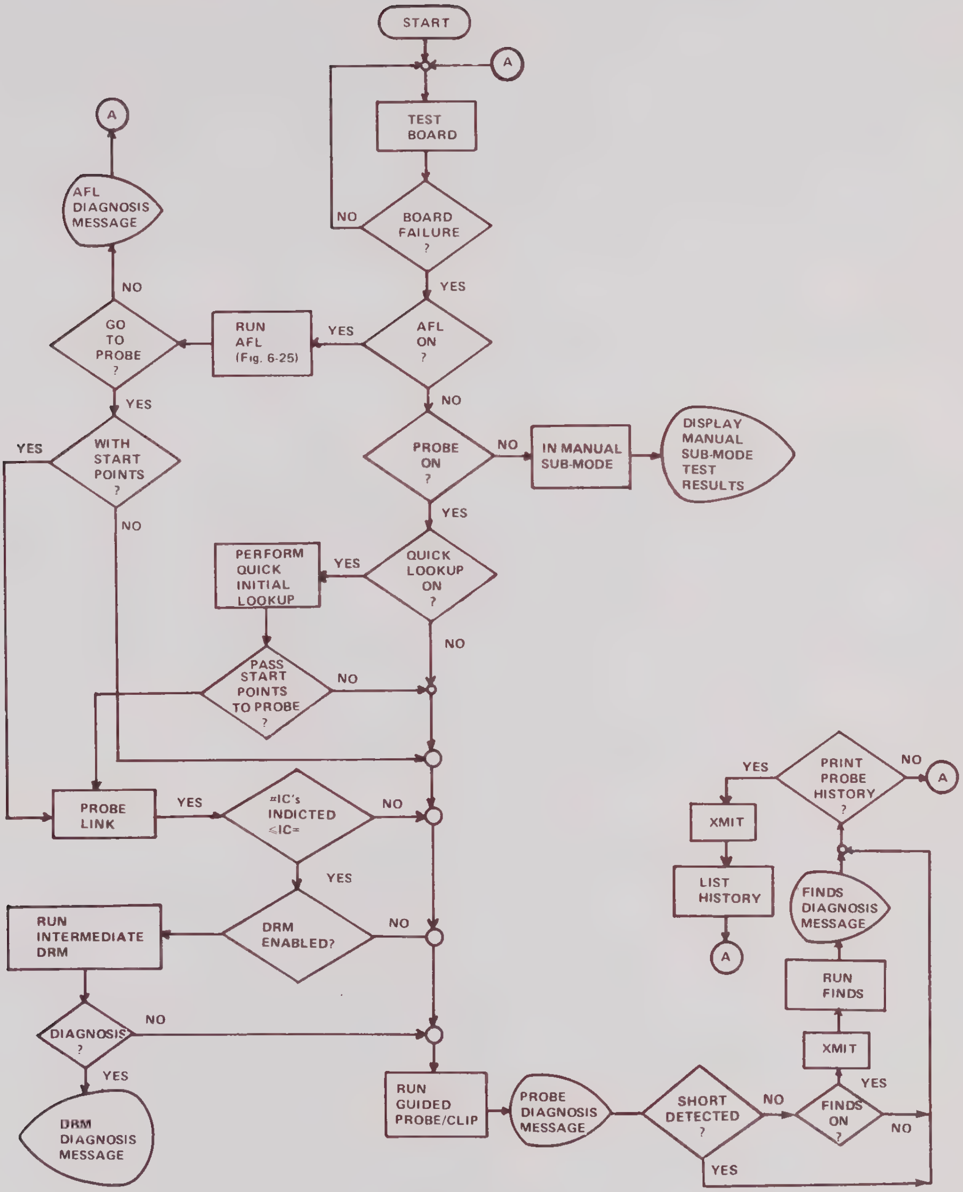

START
A
TEST BOARD
NO
BOARD FAILURE ?
YES
A
AFL DIAGNOSIS MESSAGE
NO
GO TO PROBE ?
RUN AFL (Fig. 6-25)
YES
AFL ON ?
NO
YES
YES
WITH START POINTS ?
NO
PROBE ON ?
NO
IN MANUAL SUB-MODE
DISPLAY MANUAL SUB-MODE TEST RESULTS
YES
PERFORM QUICK INITIAL LOOKUP
YES
QUICK LOOKUP ON ?
NO
PASS START POINTS TO PROBE ?
NO
YES
PRINT PROBE HISTORY ?
NO
A
XMIT
PROBE LINK
YES
#IC's INDICTED ≤IC=
NO
YES
LIST HISTORY
A
FINDS DIAGNOSIS MESSAGE
RUN INTERMEDIATE DRM
DRM ENABLED?
NO
RUN FINDS
DIAGNOSIS ?
NO
XMIT
YES
YES
DRM DIAGNOSIS MESSAGE
RUN GUIDED PROBE/CLIP
PROBE DIAGNOSIS MESSAGE
SHORT DETECTED ?
NO
FINDS ON ?
NO
YES

Figure 6-16. PROBE/FINDS Overall Flow.

4. At every point in the probing loop, the guided probe/clip checks to see if it is done, and if so, outputs the probe diagnosis message; otherwise, the probe/clipping loop continues until a diagnosis can be reached.

5. A probe History of points probed may also be output after completing the probe diagnosis by depressing the XMIT Button if the PRINT PROBE HISTORY option is ON, and FINDS is OFF. If FINDS is ON, FINDS will be run when the XMIT Button is depressed. After completing a FINDS Diagnosis, the XMIT Button can be depressed again to output the probe history.

6.5.2 Guided Probe/Clip Operation

The Guided Probe/Clip first selects its beginning probe/clip point by using the Quick Lookup and/or AFL starting points. If the "S" option is ON, however, the starting points are determined by the user starting points. If any of these points are applicable to the failure at hand, they are probed in order of priority. However, if all points probed are good, (or the "S" option is disabled), the Guided Probe/Clip will try to find a failing driver (EXTIN). If this fails, the lowest-numbered failing external will be sought. The Guided Probe/Clip then tracks back from this point.

The Guided Probe's basic algorithm is to track a failure back to its origin node by node. Starting at a bad node, it asks the operator to probe (or clip) those IC input pins which affect the output pin that is driving the bad node. If one of these inputs is bad, the algorithm directs its attention to that input pin and continues. Otherwise, it decides it has found the fault and makes a final diagnosis. The probing algorithm considers a point to be bad if its observed values fail to match its simulated values for all test steps up to and including the UUT's current lowest failing test number - the lowest failing test number being updated whenever a point with an earlier failing test is found.

Figure 6-17 is an example of a probing sequence for a faulty circuit. In addition to the node stuck type of failure of the example, the probe can diagnose broken connections between pins, failures in a feedback loop, and others.

The Guided Probe/Clip utilizes two techniques in order to assure correct operator probing: checkpointing and conditionality. The checkpointing technique involves asking the operator to probe two points on the same node. By checking that the values read by the probe are the same, the Guided Probe can check for correct probe placement and contact as well as possible broken connections. The checkpointing operation is done with a frequency as dictated by the CK= parameter. If, for example, CK=3 then at every third bad point encountered a checkpoint will be taken. CK's range is 1-7; any other value will result in a value of CK=7 being used.

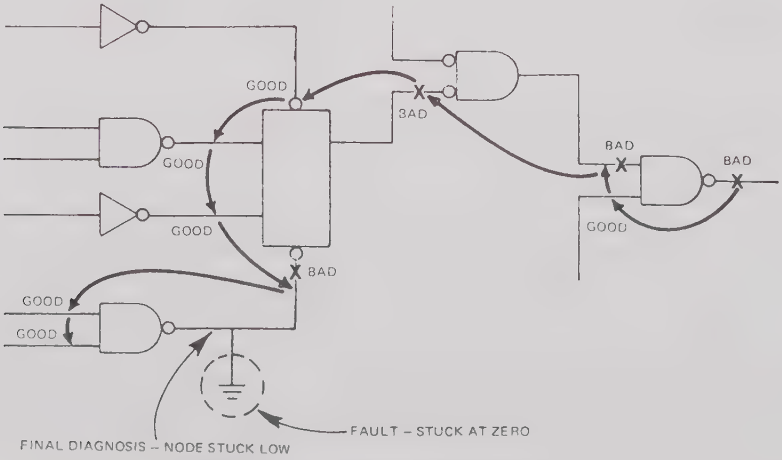


Figure 6-17. A Typical Probing Sequence.

Conditionality, the second technique used, involves an automatic evaluation of the probe placement based on the activity of the pin being probed. If a probed point never changes or changes only once during the test program's execution, then the Guided Probe considers the point "conditionally" good, bad or pass. Points that change two or more times are not considered conditional. These conditional points are accepted by the Guided Probe as correct while normal probing is under way, but while verifying a tentative diagnosis, conditional points important to the diagnosis are reprobed as a check on their correctness. This is done because it is more likely that an inactive pin was mis-probed and not noticed by the Guided Probe than that an active pin was mis-probed and went unnoticed. If CK≥3, conditional points are not reprobed.

Another important characteristic of the Guided Probe/Clip is that all points probed or clipped are remembered. Thus while following the probe's operation, it may appear that a point has been skipped - in reality the point has been previously probed or clipped and was remembered in the Guided Probe/Clip memory.

UUT's can behave intermittently due to a number of causes, among them oscillation caused by shorts and threshold region voltages caused by overloads. These UUT's could be a problem for the probe to diagnose because the Guided Probe must necessarily work by tracing back bad circuit nodes. If a UUT is intermittent, any node may be bad during one probe and good during the next. This can easily lead to an incorrect diagnosis if not handled properly. This situation is handled by reapplication of the test program to the UUT. During this multiple testing, the point being probed is watched. Only the probe values for the earliest failing test are used. This earliest occurrance is most likely to present a model situation consistent between probes thus forming the greatest chance of a correct diagnosis. The number of times that the test program is rerun is determined by the PR= parameter.

Pulses encountered while testing a UUT can easily be detected by the GR pulse-catching probe. The probe detects bad logic levels, illegal or missing pulses, and half-pulses between low and high user specified threshold levels.

The Guided Probe's normal operation can be altered both in an automatic, predetermined way and in an interactive manner at testing time. If the J option is enabled, then the jumping points specified by the user's .UJ file (User Jumping points file) are looked for during the Guided Probe's operation. If such a point is encountered, the Guided Probe uses the jump-to point specified next rather than its normal selection. Through the use of the ERROR RECOVERY control panel button and the TEC keyboard and display, the operator can also direct the Guided Probe's operation. Details of the use of this feature are described in para 6.5.4.

6.5.3 Normal Probing (With Either Probe or Clip)

The Guided Probe directs the operator as to where to place the probe or clip via the TEC display. Figure 6-18 is an example of such a display. Once the display is completed, the Guided Probe rings the TEC and teletype bells and turns on the light over the probe button.

NOTE

> The bell rings twice each time a different IC is displayed on the screen. When additional pins on the same IC are requested, the bell rings only once.

The operator should then place the probe or clip on the point indicated in the starred box of the display and depress the PROBE button on the control panel. On systems equipped with the PROBE foot-switch option, the foot switch may alternatively be depressed.

```
MESSAGE: THIS IS A PIN MESSAGE


                             ******************
                            **                **
PROBE:  ( 14 PINS )        **       E1.4       **
                            **                **
                             ******************

ICTYPE:                             7402

STATUS: CHECKPOINT BEING TAKEN

=================13-OCT-80 DISK SR PROBE HISTORY===================

LAST POINT:      AL1     DIAGNOSIS: BAD  /# 1   START (P)  (A---)
LAST BAD POINT: AL1      DIAGNOSIS: BAD  /# 1   START (P)  (A---)
ORIGINAL FAILING TEST #: 1
```

Figure 6-18. Sample Guided Probe/Clip Message.

If the IC CLIP is ON, the operator will be requested to clip a clippable IC instead of probe it. This can hasten the probing sequence significantly as data for the entire IC is obtained each time an IC is clipped. The data is then retained by the probe/clip memory and used by the guided probe on an "as needed" basis. If an IC cannot be clipped for any reason, it is only necessary to depress the XMIT Button in order to probe the IC instead of clip it.

The IC CLIP always connects to DRM CLIP B (never CLIP A). Either a 24 or 16-pin IC clip may be used in place of the probe. The 16-pin clip may be used to clip 14-pin IC's. When clipping 14-pin IC's with the 16-pin clip, pin 1 (marked) of the clip must be placed on IC pin 1. If the clip is improperly connected, a BAD CONTACT-RECLIP-USE CLIP B Message may be generated during the clipping sequence. If this message occurs, always re-check the IC CLIP connections, and the IC being clipped before depressing the PROBE Button. If the message continues, depress the XMIT Button to probe the IC instead of clip it.

NOTE

> If the message CONTINUOUS BAD CONTACT - SWITCHING TO PROBE appears, the probe button has been pushed repeatedly and failed to make contact. The system then selects the probe rather than the clip.

Various additional information is displayed on the TEC display during the probing operation. An understanding of the meaning of the various fields is useful for anyone trying to follow the probe's actions during probing.

At the top of the display, a message associated with the pin to be probed may be displayed next to the "MESSAGE:". Messages which appear there are usually directions for placement of the probe or clip. These might help the operator locate the component or give an alternate location to probe or clip. For example:

MESSAGE: THIRD RESISTOR FROM THE RIGHT

or

MESSAGE: PROBE IC3.2 INSTEAD

The operator should always look for this message before placing the probe or clip.

These messages, like diagnosis printout messages, are obtained from the .SR file. Any text up to the first "↑" (or the whole text string is no "↑" is encountered) is considered the probe message. The message can be as long as desired up to the four lines worth of space allowed on the TEC screen. Care should be taken not to write longer messages which would overwrite probe data on the screen; making future probe point requests unreadable. It should also be noted that only the first line of the message will be erased before each new probe point, thus allowing the user (by writing on the second line) to display a message which will remain permanently on the TEC screen once it is written.

The starred box in the center of the screen indicates the point to probe or clip. To the left of the box a message of the form:

```
(XX PINS)
```

occurs. This message indicates how many pins the IC or device being probed/clipped has. XX indicates the number of pins on the IC or device.

To the right of the box a message of the form:

```
INTERMITTENT BOARD BEHAVIOR
(PR=X    EXTERNAL     INTERNAL)
```

will appear if some kind of UUT intermittency is detected. The PR=X indicates the current value of the PR= (Probe Restart) parameter (X being replaced with the actual value of PR). "EXTERNAL" and/or "INTERNAL" will appear if that respective type of intermittent behavior is detected. If no driver/sensor pins fail within the CAPS section, the message "BOARD DISPLAYS GOOD BEHAVIOR" will occur.

One of a number of messages may appear next to STATUS: indicating what the Guided Probe/Clip is up to:

nothing - normal probing activity.

CHECKPOINT BEING TAKEN - a second point on the same node is being probed to double check operator probing accuracy; the operator should take extra care in placing the probe.

AFL SUGGESTED STARTING POINT - a point suggested by AFL or Quick Lookup through the probe link is being probed.

SUGGESTED STARTING POINT - a point from the user's starting point file is being probed.

FOLLOWING SUGGESTED PATH - a jumping point from the user's jumping point file was encountered and the probe is following the suggested path.

READING DRIVER/SENSOR - PLEASE DO NOTHING -the Guided Probe is automatically probing a driver or sensor pin for which no operator assistance is required.

READING NAIL - PLEASE DO NOTHING - The Guided Probe is automatically reading a NAIL (Sensor pin) for which no operator assistance is required.

PROCESSING DRIVER/SENSOR - PLEASE DO NOTHING - the guided probe is automatically processing information about a driver/sensor pin from the original test program. The test program is not rerun.

PROCESSING NAIL - PLEASE DO NOTHING - essentially the same as above, using a nail in place of a driver/sensor.

PROCESSING CLIP DATA - PLEASE DO NOTHING - the guided probe is processing data from a previous pass. No operator intervention is necessary.

REPROBING THIS POINT - as per the operator's request, a point is being reprobed or reclipped.

BAD CONTACT-RECLIP-USE CLIP B - Perhaps the IC clip is not making good contact with the IC being clipped, is incorrectly installed, or the wrong clip is being used. Check and re-connect the clip then try again. If the message continues, depress the XMIT Button to probe instead of clip.

CONTINUOUS BAD CONTACT - SWITCHING TO PROBE - continuous bad contact has been observed by clip B. The guided probe is automatically selecting the probe instead.

DRM OVERLOAD/MISMATCH - SWITCHING TO PROBE - either the DRM is overloaded at the end of the program, or there is no usable threshold on the SETUP pages. The probe is being used instead.

BACKUP IN PROGRESS (DIRECTION POSSIBLE) - the operator, through the use of the ERROR RECOVERY button, has entered the Guided Probe's backup mode. The operator may either backup one or more probed points or direct the probe's operation more explicitly.

WORKING ON TENTATIVE DIAGNOSIS - the Guided Probe has reached a point where it thinks it has the correct diagnosis, but is now in the process of double checking conditional points.

WORKING ON FINAL DIAGNOSIS - the Guided Probe is in the process of looking for broken connections, checking for possible internal (to an IC) feedback and nodal activity.

PROCESSING DIAGNOSIS - PLEASE DO NOTHING - the guided probe is processing the diagnosis and checking necessity for additional probing. No operator assistance necessary.

USER DIRECTED PROBING IN PROGRESS - the operator has entered this mode of operation. While in this mode, the following directions or error messages may appear:

ENTER NEXT POINT OR PRESS CONTINUE

UNUSABLE - WILL INDUCE FEEDBACK

SYNTAX ERROR

IC NOT FOUND

UNUSABLE - NOT AN INPUT PIN

UNUSABLE - NO-PROBE PIN

Above the line of equal signs, today's date is displayed along with the name of the CAPS section that has failed.

The two lines below the line of equals (===) describe the diagnosis associated with the last point probed and the last bad point probed, respectively. The possible diagnoses here are:

GOOD - the point probed had correct values throughout the test program.

BAD - the point probed had an incorrect value at a test less than or equal to the lowest failing test encountered thus far.

PASS - the point probed had correct values up to the lowest failing test but had an incorrect value sometime later in the test program.

After GOOD, the words BAD or PASS, "(COND)" will appear if the point probed was considered conditional by the Guided Probe. Following this are the characters "/": and # is meant to indicate "test step number". Following the "#" will be a blank for a GOOD point, the lowest failing test for a BAD point and the first incorrect test number for a PASS point. Finally, the last thing on either line is an abbreviation of the message that appeared next to "STATUS:" when the point was requested to be probed. The abbreviations and their equivalences are:

CHECKPOINT	-	CHECKPOINT BEING TAKEN
START(P)	-	Indicates a Guided Probe selected starting point
START(A)	-	Quick Lookup or AFL SUGGESTED STARTING POINT
START(U)	-	SUGGESTED START POINT
JUMP-TO	-	FOLLOWING SUGGESTED PATH
SELF-PROBE	-	READING DRIVER/SENSOR...
BACKUP	-	BACKUP IN PROGRESS
DIRECT(U)	-	USER DIRECTED PROBING POINT
TENTATIVE	-	WORKING ON FINAL DIAGNOSIS
FINAL	-	WORKING ON FINAL DIAGNOSIS

NOTE

Refer to Para. 6.5.6 for Guided Probe history format and codes

The number following ORIGINAL FAILING TEST #: is the test number at which the board failed when first tested. A "*" following this number indicates that the original failure was at a non-digital test.

Finally, on the very last line of the screen the text "REPROBE PENDING" or "BACKUP PENDING" will appear as soon as the ERROR RECOVERY button is depressed and will disappear as soon as some relevant action has been taken by the Guided Probe/Clip.

6.5.4 Testing With The GenRad Pulse-Catching Probe

Normal probing with either the X-1 passive probe or the GenRad pulse-catching probe is described in para 6.5.3. When the GenRad pulse-catching probe is being used, the following information will also be of value:

1. The pulse-catching probe records the occurrence of a pulse, but not the number of pulses in a pulse train or the waveform.

2. If a pulse is detected, the occurrence of a half-pulse at that probed point cannot be detected.

3. The recording of results for each test step begins and ends under the following set of conditions when using the pulse-catching logic probe:

   a. Recording begins whenever new states are applied to the board via a SET LGC or SET PULSE command.

   b. Recording ends after completing a TEST LGC command. Recording may also begin when a TEST LGC command is processed if it is not already in progress because of a prior SET LGC or SET PULSE command.

   It is therefore possible to extend the recording of probe results over a test step boundary, if desired, through use of SET LGC or SET PULSE commands in the test program. When the recording of probe results begins because of a SET LGC or SET PULSE command, it will not end until the next TEST LGC command has been completed.

4. If a BROKEN CONNECTION probe diagnosis message occurs while probing an EXTernal pin that is being driven by an IC, always check to determine if the CAPS/APG simulator has modeled an incorrect pulse that the probe cannot detect. Under these conditions, the FAILING TEST number reported by CAPS/APG may not be the lowest failing test number involved. An analysis of the Good Board Simulation Results (PRETTY) listing should be made at this point to determine the correct range of test steps involved and whether the simulator has modeled the range correctly. If a false simulation has been detected, the NOPROBE command can be assigned to the node involved or the .UD file can be used to ignore any pulse errors that occur on the node.

5. Because of the extreme sensitivity of the pulse-catching logic probe, stray pulses, half-pulses, or attenuated pulses may be detected on some boards that cannot be accounted for logically. Some of these stray pulses may also be present on a known good board without affecting the board's operation, and therefore can normally be disregarded by the user. As a general rule, the following procedures should be followed to eliminate the need for excessive board probing:

   a. Set-up the probing sequence in advance by probing nodes on a known bood board first. If stray pulses are detected on the known good board, and an analysis of the board indicates that it is functioning properly, the pulses can be disregarded when testing other boards.

   b. As a general rule, always disregard half-pulses. If an analysis of the known good board indicates that half-pulses may cause problems at specific points, these points may be selectively enabled while all other points are disregarded.

   c. It is always a good idea to check outputs of the known good board against the CAPS/APG simulated outputs through analysis of the Good Board Simulation Results (PRETTY), and resolve any differences that exist. Differences may be caused by the way that 1792A/B or 1795 Driver/Sensor

Pins operate. The CAPS/APG simulator assumes that all external test steps change state at the same time, however, the pins are actually driven twelve at a time with several micro-seconds lag between the first group of twelve and the next group. If Inputs to an IC must be changed at the same time, and the Inputs are not assigned to the same set of twelve pins, the time lag may cause the simulation results to be in error or for simulated race conditions to occur at the IC's outputs.

d. Once the CAPS/APG simulation results and known good board are in agreement, the probe can be used to probe internal board nodes that are determined to have problems (unwanted pulses, bad levels, etc.) as indicated by studying the PRETTY printout. It is always a good idea to set-up the board problem areas in advance so that the pulse-catching logic probe can be used to isolate faults on the problem nodes rather than attempt to search the entire board. Probe starting and jumping point files can be used to establish starting locations once these locations have been determined.

6. Noise may be detected on external board pins that are connected to sensor pins if the external pins are being driven by "open collector" logic. This type of noise occurs because the "open collector" outputs do not have pull-up resistors and the pulse-catching probe detects very small changes across threshold boundaries. Noise of this type can normally be disregarded if it does not affect board operation or pull-up resistors can be added to the circuit.

7. Noise may also be detected if Driver and Sensor pin wiring is not properly isolated in the Device Adaptor. Running Driver and Sensor pin wiring close together or using excessively long leads in the Device Adaptor can cause noise on some pins. Such noise may not be of sufficient magnitude to trigger the test system Sensors, however, the sensitive pulse-catching probe will detect it easily. Depending on the level of fault isolation desired, noise of this type can either be eliminated by correcting the Device Adaptor wiring or disregarded through use of the .UD file.

8. When preparing a .UD file, it is normally easier to prepare an ENABLE file rather than an IGNORE file. However, Nodal Verification prepares only IGNORE Type .UD files. The ENABLE file allows selective pins to be enabled while ignoring all others. It is thus possible to limit the activity of the pulse-catching logic probe to those pins and nodes that require probing for specific pulse errors while ignoring those discrepancies that do not affect a board's operation. The .UD file may be used whenever the D option has been enabled. Procedures for preparing a .UD file are contained in para 5.2.7.

9. The following pulse-catching probe error messages may occur in the DIAGNOSIS Mode:

```
PROBE READS BOTH HIGH AND LOW LEVEL -
PLEASE CHECK THRESHOLD SETUP
```

This message could indicate a possible probe hardware failure or perhaps incorrect threshold values have been assigned by the SETUP Mode.

```
IMPOSSIBLE CONDITIONS EXIST -
PLEASE CHECK LOGIC PROBE HARDWARE
```

This message indicates a possible probe hardware failure. Threshold, half-pulse, or level check data transmitted from the probe hardware is unusable.

6.5.5 Reprobing, Backup, and User Direction

The ERROR RECOVERY Button is a multiple function button that allows an operator to run in either a BACKUP PENDING or REPROBE PENDING probe/clip mode. A change from one mode to the other occurs when the ERROR RECOVERY Button is depressed the correct number of times.

The REPROBE PENDING mode enables an operator to re-probe the point currently being probed again. If for example, the probe has inadvertently slipped off the point being probed (or perhaps the wrong IC has just been probed), a re-probe of that point can be accomplished by depressing the ERROR RECOVERY Button once (depressing the button twice will cause the BACKUP PENDING mode to be entered). If an operator is able to do this before the next point to be probed is displayed (something not as difficult as it sounds), a REPROBING THIS POINT message will be displayed and the same point can again be probed.

The BACKUP PENDING probing mode enables an operator to:

1. Back-up and Re-Probe the last probed point again (or continue to back-up until the correct re-probe point is reached. If a mistake is noted prior to actuating the PROBE button to probe the current point, an operator may back-up one position (or a series of positions) by depressing the ERROR RECOVERY Button twice (depressing the button three times at this point will result in a return to the REPROBE PENDING mode). When the BACKUP PENDING message is displayed, an operator must proceed as follows:

   a. Probe the point currently requested. This step is important. Once the current point has been probed, the backup mode will be entered at the earliest opportunity (in most cases immediately), however, if the probe is checkpointing or working on a tentative or final diagnosis, these operations must be completed before the backup request can be honored. The BACKUP PENDING message will continue to flash until the request is honored, at which time a BACKUP IN PROGRESS (DIRECTION POSSIBLE) message will be displayed.

   b. When the BACKUP IN PROGRESS (DIRECTION POSSIBLE) message is displayed, depress the ERROR RECOVERY button once to back-up one probe point. The new point to be probed will be displayed. An operator may elect to continue backing up in sequence at this time by continuing to depress the ERROR RECOVERY button until the desired point to be probed is displayed.

c. Next, depress the CONTINUE button and then re-probe the point listed. Please note that the probe backs-up along a logical path and includes only those points that lead directly from the starting point to the present point. It is therefore not possible to re-probe those points that lie along tangential paths. In addition, once all of the logical points have been passed over, the probe will automatically exit the backup mode and continue along whatever path it was working on after stating that it has done so when the BACKUP IN PROGRESS Message is removed from the display screen.

2. Direct the probe or clip to a specific point to be probed (a user directed probe point). An operator may direct the probe to look at any specific point or clip a specific IC. The guided probe will then ask the operator to probe (or clip) the requested point and it will display a diagnosis for that point. A powerful tool has thus been created by utilizing the guided probe's ability to compare actual values at a given point with expected values in combination with an opertor's experience and intuition. When the BACKUP IN PROGRESS (DIRECTION POSSIBLE) message is displayed, an operator should proceed as follows:

a. List the point to be probed via the TEC keyboard followed by an XMIT. For example, the entry

IC2.3 (and XMIT)

will request that pin 3 of IC2 be probed. The only restrictions on what pins may be requested are that they be probeable input pins which have not yet been probed. The IC may also be clipped (instead of probed). If the IC is clipped, the new data will be used for future references to that IC.

b. Once the pin has been entered, it will either be accepted or rejected. If it is rejected, an error message will appear on the STATUS message line. If it is accepted, then the PROBE light will be turned ON and the operator should then probe/clip the point indicated.

c. If the IC CLIP option is ON, and it is desired to delete data saved from a previous clipping, the IC data may be listed with an asterisk (*) via the TEC keyboard followed by an XMIT. For example,

*IC2 (and XMIT)

would cause the guided probe/clip to remove all data concerning IC2 from memory and then automatically start the clipping sequence again at the correct location. Note that an IC does not have to be clipped in order to delete it from memory. If IC2 is deleted via the asterisk (*) input, then all data covering IC2 and any of it's pins will be deleted at the same time. It is only necessary that the IC CLIP option be ON in order to use the delete feature. In certain cases, however, if the first (or anchor) IC is deleted, the probe/clip may restart the clipping sequence from the beginning.

d. After the point indicated has been probed and judged, further points to be probed can be listed via the TEC keyboard and the procedure again repeated. When it is desired to return control to the guided probe, depress the CONTINUE button. The guided probe always attempts to take advantage of the points that an operator has asked to be probed by continuing on from the last bad point that has been probed. If there is no such point, then the guided probe will proceed from where it left off when control was taken from it.

The following general notes are appropriate with respect to the CAPS Probe control panel lights and buttons:

> The only buttons and lights that have any special meaning to the probe software are the PROBE Button and light, the ERROR RECOVERY Button, the CONTINUE Button, and the PAUSE light. Each of these operate in the following manner:
>
> a. When the PROBE light is on, it is a request *to probe* the point indicated. Probe *only* when the light is *on*, but *always probe* when the light is *on*.
>
> b. When the PAUSE light is on, it signifies that the CONTINUE Button may be depressed to proceed. In the BACKUP mode, CONTINUE means that this is the point to be re-probed. If operating under user direction, CONTINUE means that control should be returned to the probe.

6.5.6 Guided Probe History

The Guided Probe/Clip, during its operation, saves a readable history of the points probed/clipped. This history, an example of which is shown in Figure 6-19, can be output to the listing device (LI on the FILE USAGE Page) *after* the probe diagnosis message is output. If FINDS is OFF, *and* the PRINT PROBE HISTORY option is ON, the probe history can be output by depressing the XMIT Button. If FINDS is ON, FINDS will be run when the XMIT Button is depressed. *After* the FINDS diagnostic message is output, the probe history can then be output by again depressing the XMIT Button. The probe history may be of value in tracing simulation mismatches and collecting statistics in order to determine better starting points.

When a final PROBE/FINDS diagnosis appears on the TEC screen, the system will wait for one of several events. If XMIT is entered on the TEC keyboard or any character is entered on the TTY (except ↑C), then the system will check the "print probe history" option and print the history if it is enabled. If the START or ABORT button is hit, however, the printing of the probe history is suppressed and new board testing is initiated. In this way, a probe history can be optionally printed out or not depending on the operator's discretion.

```
13-OCT-80 DISK SR PROBE HISTORY
AF1;        BAD        /# 8    START (P)     (A---)
E8.1;       GOOD       /#                    (A---)
E8.2;       GOOD       /#                    (A---) -N
E8.3;       GOOD       /#                    (A---)
E8.4;       BAD        /# 2                  (SL--)
E11.9;      GOOD       /#                    (A---) -C
E11.8;      BAD        /# 2    FINAL         (SL--) -C
```

Figure 6-19. Probe History.

> (SL) Characters in Parentheses. This is a two-character probe/clip results code that can be used to determine the type of discrepancy that has been detected by the probe. For example, the code (--IP) indicates that an illegal pulse has been detected. Table 6-1 lists the complete set of coded characters that can occur. If the GR pulse-catching probe is *not* being used, the code *will* still occur, but will reflect the capability of the measuring device. For example, the X-1 probe or clip may detect stuck bad levels (SH--), active High Levels (AH--), no errors (A---), etc., but cannot detect oscillations, illegal or missing pulses, etc.

The probe history itself consists of one line of text per point probed. The line has the same general contents as the "LAST POINT PROBED:" message displaye on the TEC screen while probing (see Figure 6-18). In addition, the following information may appear:

-C Clip
-N Nail

Table 6-1

PROBE/CLIP RESULTS CODE

Level	Pulse	The Probe/Clip Has Detected
A	-	No errors
	HP	Half-Pulse
	MP	Missing pulse
	-	Attenuated pulse (missing and half-pulse)
	IP	Illegal pulse
	OS	Both high and low thresholds have been detected (probably caused by an oscillation or threshold overlapping).
AH	-	Active high level
	HP	Half-Pulse ending high
	MP	Missing pulse ending high
	-	Attenuated pulse ending high
	IP	Illegal pulse ending high
AL	-	Active low level
	HP	Half-Pulse ending low
	MP	Missing pulse ending low
	-	Attenuated pulse ending low
	IP	Illegal pulse ending low
AM	-	Active mid-range level
	HP	Half-Pulse ending mid-range
	MP	Missing pulse ending mid-range
	-	Attenuated pulse ending mid-range
	IP	Illegal pulse ending mid-range
S	-	Stuck good level
SH	-	Stuck bad high level
	MP	Stuck bad high level and missing pulse
SL	-	Stuck bad low level
	MP	Stuck bad low level and missing pulse
SM	-	Stuck bad mid-range level
	MP	Stuck bad mid-range level and missing pulse

6.6 FINDS TESTING

6.6.1 Introduction

FINDS testing (see Figure 6-20) is based on the summary diagnosis supplied b the Guided Probe. The FINDS operation performs its diagnosis to eliminate each type of fault until the actual fault is encountered. The types of faults detected by FINDS are listed below in the order that they are investigated:

Shorts/opens between an IC pin and power or ground
Shorts/opens between adjacent IC pins
Faulty IC's
Shorts between tracks on an etch board.

FINDS first performs the "power short test". By measuring the continuity between the IC pin connected to the faulty node and each pin specified as power or ground on that IC, FINDS attempts to isolate the fault to an incorrect short or open between the points measured.

The second test performed by FINDS is the "adjacent pin short test". This test measures the continuity between the faulty node and every pin of each IC connected to the faulty node, attempting to find an incorrect short or open.

The third test performed by FINDS is the "μvoltage test", which verifies that each IC connected to the faulty node is functioning properly. This is done by measuring the voltage at each driving IC pin connected to the faulty node. If the summary diagnosis supplied by the guided probe is a feedback diagnosis, the μvoltage test is not performed. The μvoltage test is not performed within 1796 FAST sections.

The last test performed by FINDS is the "track short test". This test is performed under operator direction and the continuity is measured between tracks on the etched board. FINDS determines whether the continuity is correct or incorrect. At any time during the "track short test" the operator may initiate the FINDS operation on a new node in a feedback loop. This is done at the operator's discretion.

FINDS functions only following the guided probe operation. Once the guided probe completes its operation, if the display intermediate guide probe diagnosis (P) option is on its diagnosis is displayed before the system links to FINDS, otherwise it links directly to FINDS.

NOTE

> Probe History can not be displayed after the intermediate probe diagnosis-only after the final diagnosis.

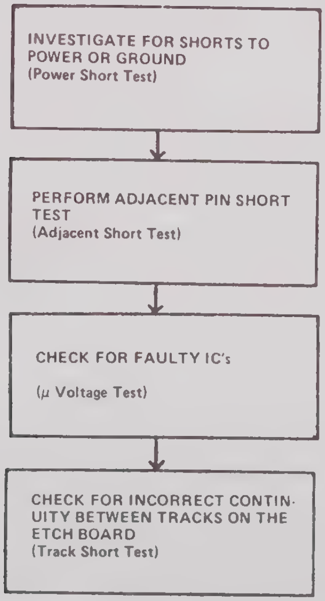

INVESTIGATE FOR SHORTS TO POWER OR GROUND
(Power Short Test)
PERFORM ADJACENT PIN SHORT TEST
(Adjacent Short Test)
CHECK FOR FAULTY IC's
(μ Voltage Test)
CHECK FOR INCORRECT CONTINUITY BETWEEN TRACKS ON THE ETCH BOARD
(Track Short Test)

Figure 6-20. Method of Fault Analysis for Bad Nodes.

6.6.2 FINDS Displays

Various information is displayed on the TEC terminal during the FINDS operation (see Figure 6-21).At the top left of the display screen there are two message areas. They are MSSGE1: and MSSGE2:. These messages are to assist the operator and relate to the point probed or IC clipped. (MSSGE1: pertains to the first or only point to be probed or IC clipped and MSSGE2: pertains to the second point probed or IC to be clipped.) These messages are created during the .SR file generation or Z Sub-Mode of PREPARE.

These messages are the same messages used in the guided probe operation. The next heading is PROBE:, which lists the point to be probed, IC to be clipped, or other directions for the operator (see para. 6.6.3.4). The information of what and how to probe is contained in a starred box. If the IC has other than 14 pins, the correct number of pins is displayed between PROBE: and the corresponding pin/IC entry in the starred box. Under the PROBE: heading is the heading STATUS:. The status information permits the user to see exactly what FINDS is attempting to accomplish or what user action has been performed. The status messages are listed in Table 6-2. To the right of the starred box SUSPECTED TRACK SHORT is listed when FINDS is performing the adjacent track short test.

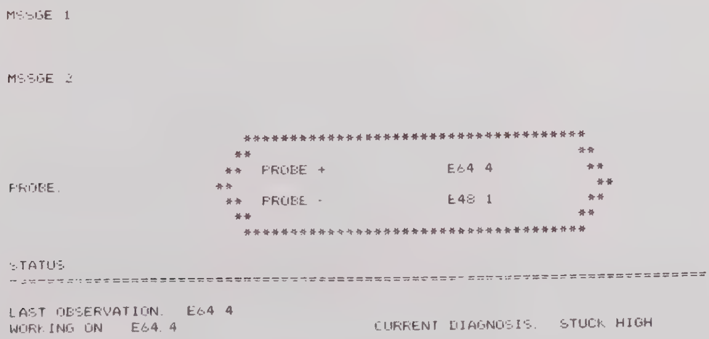
The display is now divided by a line of equal signs. Below this line are additional headings; the first is LAST OBSERVATION:. Here FINDS lists the last point clipped or probed for reference during error recovery. The second heading is WORKING ON:, which lists the IC or pin currently being clipped or probed. The last entry is CURRENT DIAGNOSIS:, which lists the nodal diagnosis from the Guided Probe.

During the track short test, an additional entry appears beneath the line of equal signs. It is the nodel clip/probe placement reference message and its intended to remind/inform the operator where contact is to be made to the node while searching for track shorts.

One final message area is at the bottom of the screen and is used to indicate the error recovery status, either "pending" or "in progress."

6.6.3 FINDS Test Flow

Reference Figure 6-22 for the following discussion.



```
MSSGE 1


MSSGE 2


                           *************************************
                          **                                    **
                         **  PROBE +            E64 4            **
PROBE.                  **                                        **
                         **  PROBE -            E48 1            **
                          **                                    **
                           *************************************

STATUS
==============================================================================

LAST OBSERVATION.  E64 4
WORKING ON   E64. 4                      CURRENT DIAGNOSIS.  STUCK HIGH
```

Figure 6-21. Typical FINDS Display.

Table 6-2

FINDS STATUS MESSAGES

MESSAGE	EXPLANATION
PROCESSING BROKEN CONNECTION - DON'T REMOVE CLIP B.	Performing continuity check - both clips/probes must be in position
RECLIPPING DUE TO BAD CONTACT	FINDS is unable to verify CLIP B is making prober contact to the IC. Check placement and repeat test.
REPEATING MEASUREMENT DUE TO INCONSISTENCY	Diagnosis is inconsistent-repeat process
WORKING ON FINAL DIAGNOSIS	Indicates to the operator that a diagnosis has been determined, but is yet to be verified-the point(s) must be reclipped/reprobed to verify the diagnosis
INVALID COMMAND, TRY AGAIN	Operator specified a chip to be clipped that can not be clipped or other invalid command - see instructions in starred box for valid commands
TESTING CLIP/PROBE PLACEMENT ON NODE	FINDS is performing contactivity testing to the nodal IC for the suspected track short test
NODAL PLACEMENT NOT VERIFIED	Nodal placement for the track short test was not verified. Instructions to repeat the test are displayed in the starred box
NODAL CLIP/PROBE PLACEMENT CAN'T BE VERIFIED	Due to certain limitations (too many nodal pins are unclippable or unprobable) placement of clip or probe can not be verified. (Appears during tracks shorts.)
CONTINUITY DETECTED	There is a short between two points probed during track shorts. Instructions to identify the clipped/probed point are displayed in the starred box
CAN'T VERIFY CLIP PLACEMENT	This applies to clip B during tracks shorts and indicates the system can not assume clip B is making good contact to an IC since no contactivity is detected at one of the four corners. It is necessary to identify the clipped IC so that the system can properly check it for shorts
ARE YOU SURE	IC/pin identified during the track short test is either NOPROBED or NOCLIPPED. Enter YES to test the IC/pin
REMOVE CLIP WHILE PROBING	Remove clip B while in "probe mode" phase of the track short test - no test is performed until clip B is removed!

6.6.3.1 Diagnosing Opens or Shorts to Power or Ground

FINDS first determines whether the guided probe diagnosed a bad node or a broken connection. If the node is bad then FINDS determines if it is stuck. If the node is stuck, FINDS checks the continuity between an IC pin connected to the faulty node and each pin specified as power or ground on that IC in the library. It also checks continuity between the same IC pin and any pin specified as power during percent processing. With this procedure FINDS is able to determine if there is incorrect continuity which signifies either a broken connection or a short. If such a fault is detected FINDS terminates processing and displays the diagnosis.

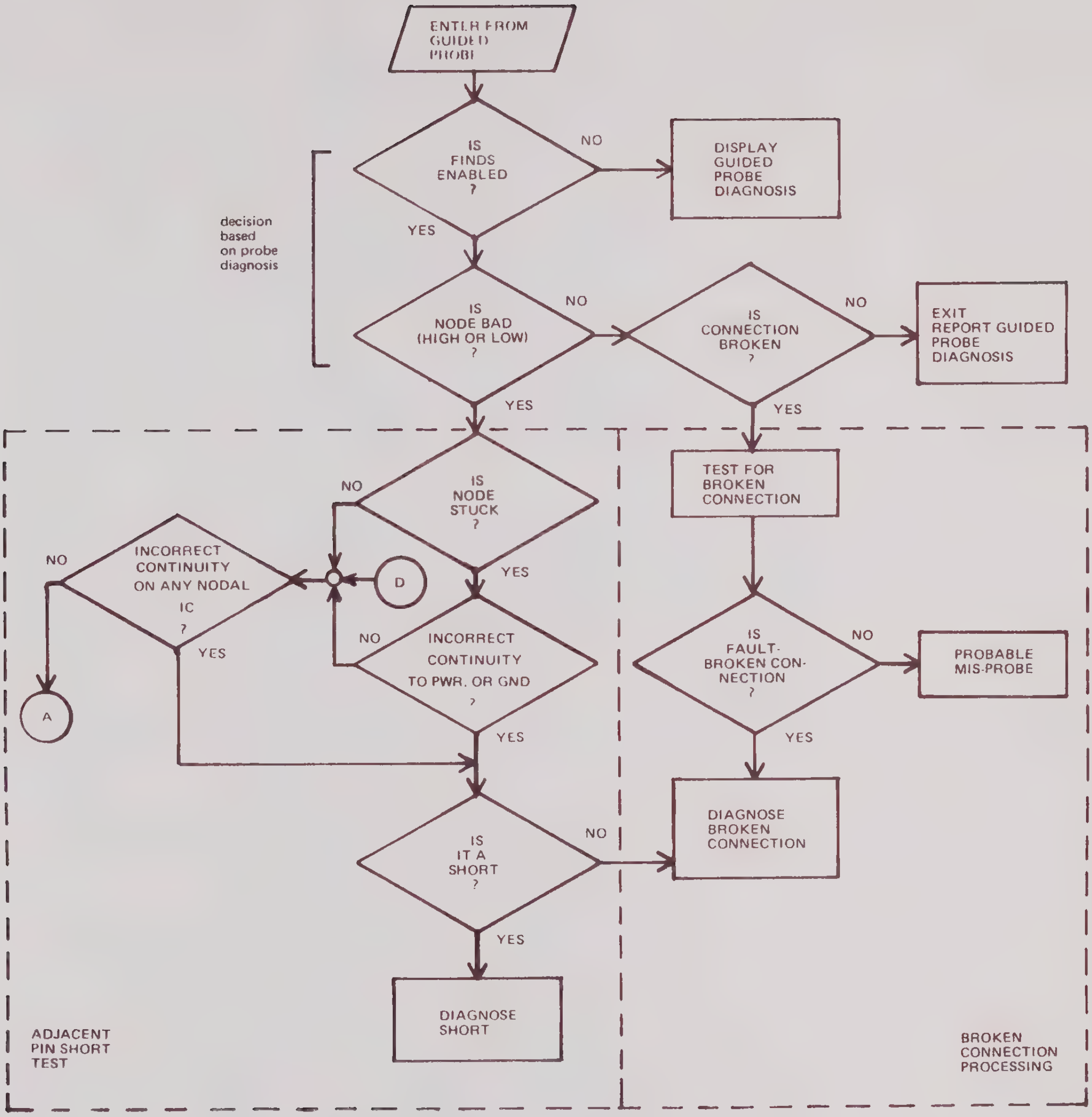

ENTER FROM GUIDED PROBE
IS FINDS ENABLED ?
NO
DISPLAY GUIDED PROBE DIAGNOSIS
YES
decision based on probe diagnosis
IS NODE BAD (HIGH OR LOW) ?
NO
IS CONNECTION BROKEN ?
NO
EXIT REPORT GUIDED PROBE DIAGNOSIS
YES
YES
IS NODE STUCK ?
NO
TEST FOR BROKEN CONNECTION
NO
INCORRECT CONTINUITY ON ANY NODAL IC ?
D
YES
YES
NO
INCORRECT CONTINUITY TO PWR. OR GND ?
IS FAULT-BROKEN CONNECTION ?
NO
PROBABLE MIS-PROBE
A
YES
YES
IS IT A SHORT ?
NO
DIAGNOSE BROKEN CONNECTION
YES
DIAGNOSE SHORT
ADJACENT PIN SHORT TEST
BROKEN CONNECTION PROCESSING

Figure 6-22. FINDS Flow Chart (Sheet 1 of 3).

6.6.3.2 Adjacent Pin Shorts

If FINDS determines that there is correct continuity between the IC pin connected to the faulty node and power and ground it then checks continuity between the faulty node and all pins on each IC connected to the faulty node. If incorrect continuity is detected, FINDS determines if the incorrect continuity is a short or broken connection and displays the diagnosis on the TEC terminal.

6.6.3.3 Testing For Faulty IC's

After FINDS determines that a fault does not exist between pins on each nodal IC it checks to verify that each nodal IC is functioning properly. This is accomplished by measuring the voltage at each driving IC pin connected to the node.

FINDS first looks for a feedback diagnosis. If it is a feedback diagnosis, the μvoltage testing is not performed. If the correct state of the faulty node is low, bu

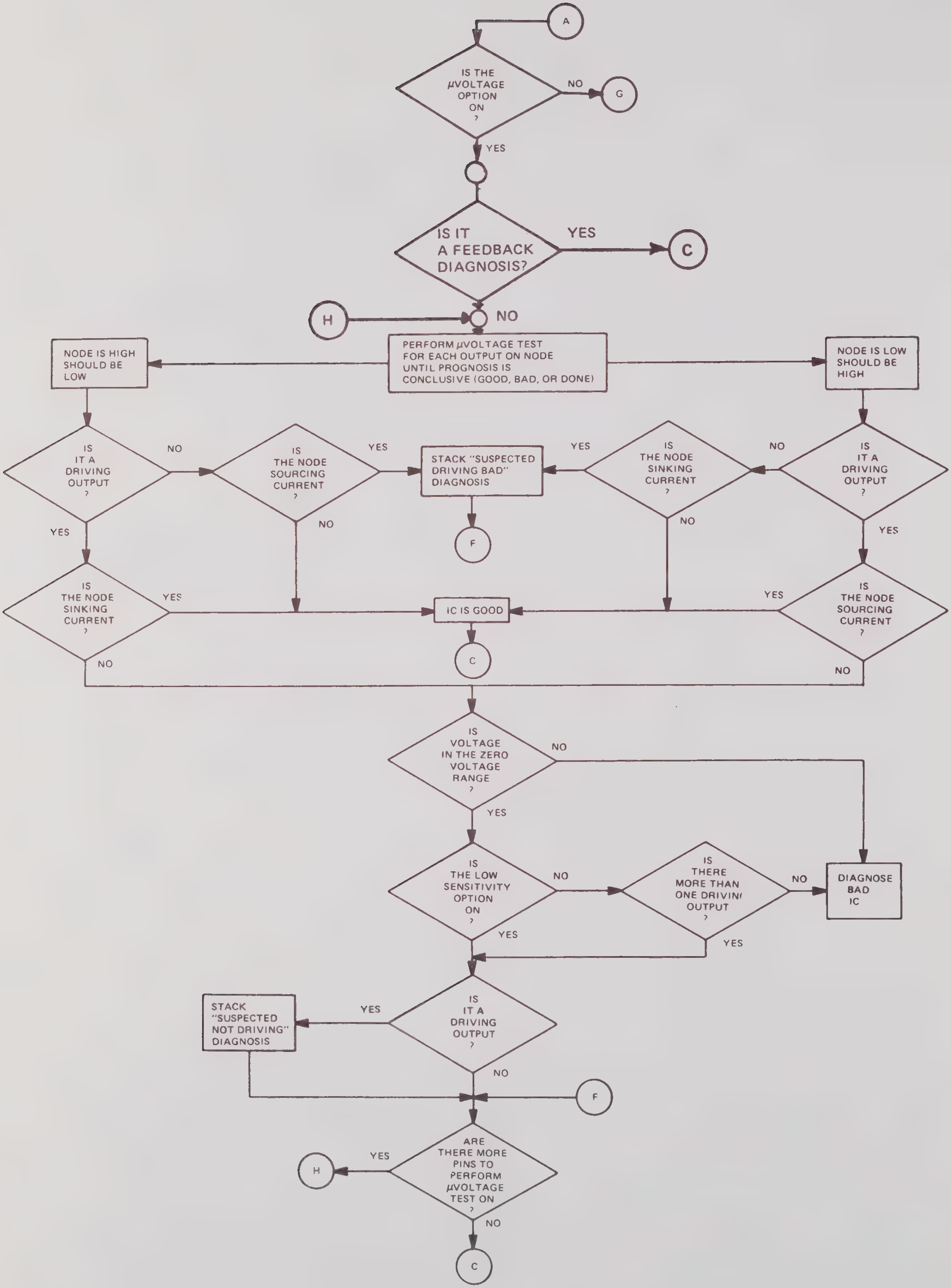

A
IS THE μVOLTAGE OPTION ON ?
NO
G
YES
IS IT A FEEDBACK DIAGNOSIS?
YES
C
H
NO
PERFORM μVOLTAGE TEST FOR EACH OUTPUT ON NODE UNTIL PROGNOSIS IS CONCLUSIVE (GOOD, BAD, OR DONE)
NODE IS HIGH SHOULD BE LOW
NODE IS LOW SHOULD BE HIGH
IS IT A DRIVING OUTPUT ?
NO
IS THE NODE SOURCING CURRENT ?
YES
STACK "SUSPECTED DRIVING BAD" DIAGNOSIS
YES
IS THE NODE SINKING CURRENT ?
NO
IS IT A DRIVING OUTPUT ?
YES
NO
F
NO
YES
YES
IS THE NODE SINKING CURRENT ?
YES
IC IS GOOD
YES
IS THE NODE SOURCING CURRENT ?
NO
C
NO
IS VOLTAGE IN THE ZERO VOLTAGE RANGE ?
NO
YES
IS THE LOW SENSITIVITY OPTION ON ?
NO
IS THERE MORE THAN ONE DRIVIN(OUTPUT ?
NO
DIAGNOSE BAD IC
YES
YES
STACK "SUSPECTED NOT DRIVING" DIAGNOSIS
YES
IS IT A DRIVING OUTPUT ?
NO
F
ARE THERE MORE PINS TO PERFORM μVOLTAGE TEST ON ?
YES
H
NO
C

Figure 6-22. FINDS Flow Chart (Sheet 2 of 3).

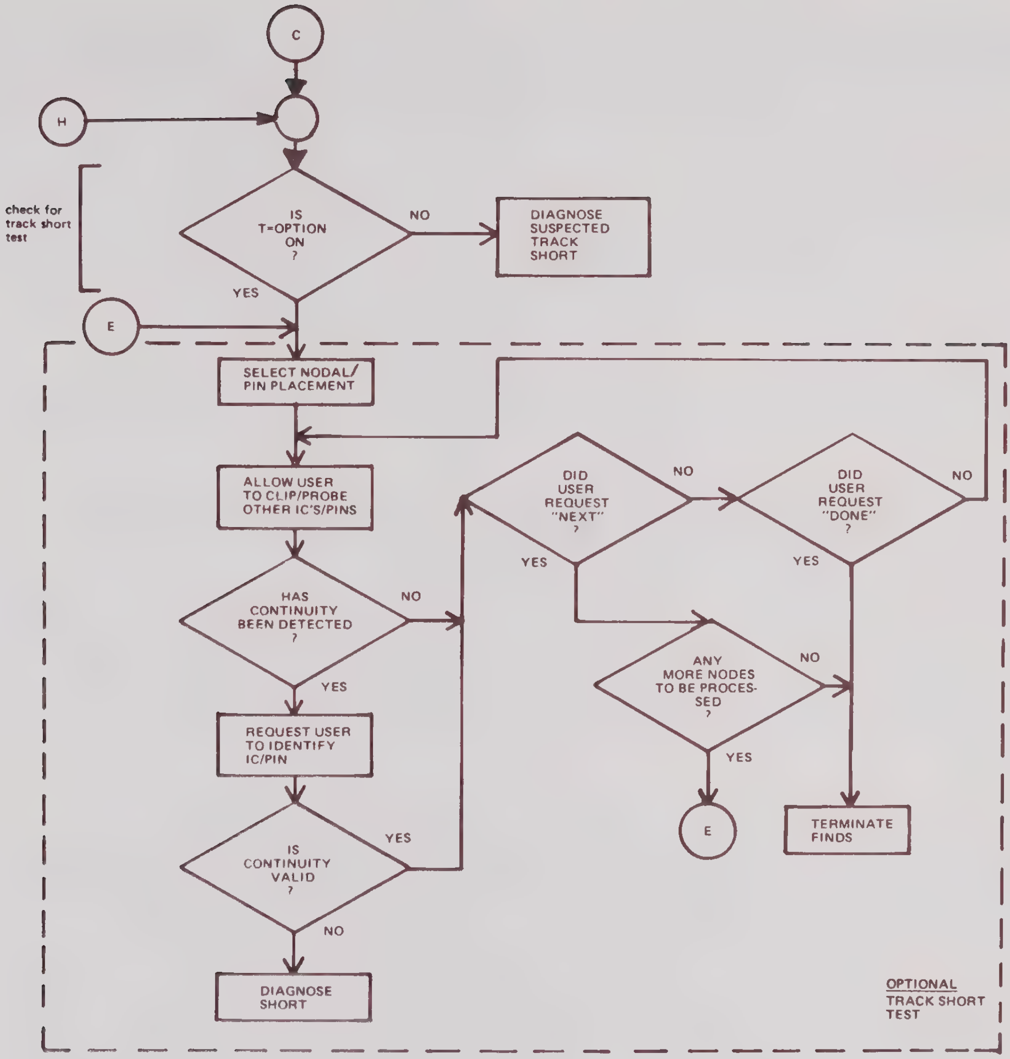

C
H
check for track short test
IS T-OPTION ON ?
NO
DIAGNOSE SUSPECTED TRACK SHORT
YES
E
SELECT NODAL/ PIN PLACEMENT
ALLOW USER TO CLIP/PROBE OTHER IC'S/PINS
DID USER REQUEST "NEXT" ?
NO
DID USER REQUEST "DONE" ?
NO
YES
YES
HAS CONTINUITY BEEN DETECTED ?
NO
YES
ANY MORE NODES TO BE PROCESSED ?
NO
YES
REQUEST USER TO IDENTIFY IC/PIN
E
TERMINATE FINDS
IS CONTINUITY VALID ?
YES
NO
DIAGNOSE SHORT
OPTIONAL TRACK SHORT TEST

Figure 6-22. FINDS Flow Chart (Sheet 3 of 3).

its present state is high then each driving output pin is checked to determine if it is sourcing (voltage drop greater than 10μ volts) or sinking (voltage drop less than -10μ volts) current. If it is not a driving output and not sourcing current then tha IC is good. The IC is also good if it is a driving output and it is sinking current. If it is not a driving output but it is sourcing current then FINDS stacks the diagnosis "suspected driving bad". A diagnosis that is stacked is displayed along with the original probe diagnosis if FINDS can not refine the probe diagnosis. If it is a driving output and it is neither sourcing or sinking current then FINDS determines whether or not the voltage is in the zero volts range.

When the correct state of the faulty node is high but its present state is low FINDS performs a similar type operation as explained in the previous paragraph. The differences are apparent in Figure 6-22 (sheet 2 of 3).

If the low sensitivity option is off and there is not more than one driving output then again the IC is diagnosed as bad. When the voltage is in the zero volts range and low sensitivity is on, FINDS determines if it is a driving output. If it is then FINDS stacks a "suspected not driving" diagnosis and then determines if there are more pins to be tested with the μ voltmeter. If there are more pins to be processed, FINDS continues to check each pin. If this was the last pin on the node then FINDS determines the state of the extended track short detection option. If it is off FINDS terminates with a suspected track short diagnosis.

NOTE

> If for any reason the operator does not want to perform the μ-voltage testing he may depress the XMIT key and the μ-voltage testing is not performed.

6.6.3.4 Processing Suspected Track Short

NOTE

> When processing suspected track shorts the message SUSPECTED TRACK SHORT is always displayed on the terminal screen.

If the suspected track short option is on (T=ON), FINDS allows the operator at his discretion to check for a suspected track short. First, however, the system must select a nodal point to be clipped or probed (by clip A, or either probe) while the operator uses clip B or a probe to test for shorts to the other IC/IC pins on the board.

Since CLIP A is used to clip the nodal IC, special precautions are required to verify that CLIP A is properly positioned. Contactivity of CLIP A to the IC can not be checked as a single operation. It is necessary to place CLIP A on the IC (IC PIN 1 to CLIP PIN 1) and probe a second nodal point to verify that nodal contact is achieved with clip A.

If continuity can not be verified the user has the option to either retry the verification by pressing the XMIT key or perform the suspected track short testing without verification by pressing CONTINUE.

If at any time during the track short test operation, clip A is accidently moved it can be replaced and its contact checked by entering *V.

If no nodal point is clippable it is necessary to probe the node for each track short test - while clipping or probing the test point. If the node is to be probed the system attempts to verify the operator's understanding of the nodal position by probing two points on the node simultaneously.

If only one nodal point is clippable/probeable, the system indicates that nodal placement cannot be verified.

During the entire track short process, the nodal placement is displayed as "NODAL CLIP/PROBE PLACEMENT:"

Once nodal contact is determined, FINDS displays the appropriate instruction and waits for an operator's action. At this time the operator can use clip B to clip any and all IC's on the board, depressing the probe button for each test. It is recommended that a standard pattern be used for this process such as starting at one corner of the board and progressing vertically or horizontally back and forth through the entire board.

If continuity to the node is detected the system requests the operator to identify the IC or pin clipped or probed. Given this information the system requests the operator to clip/probe as necessary. If continuity is expected and verified the operator can move on to the next test. Otherwise the system will enter "final diagnosis."

Other operations valid during track short test are to: directly identify an IC or pin to be clipped/probed, change the mode from clipping to probing the test IC and vice versa as well as entering *NEXT to specify a new node on a feedback loop or entering *DONE to terminate the FINDS operation.

*P is entered to specify further test points are to be probed rather than clipped with clip B. During probing of a node, clip B must be removed. If it is not, no test is performed and a message to remove it is displayed and a warning bell is rung. Probing test points should be performed in the same systematic approach as clipping

*C is entered to specify return to Clip Mode. The current mode (clip or probe) is indicated by the track short instruction. When clipping a node the message is:

CLIP NEXT POINT OR, ENTER IC (#) TO BE TESTED OR"*VERIFY","*CLIP","*PROBE",
"*NEXT" OR "*DONE"

When Probing a node the message is:

PROBE NEXT POINT OR ENTER IC (.#) TO BE TESTED OR "*VERIFY," "*CLIP,"
"*PROBE," "*NEXT," OR "*DONE."

If clip B is used contactivity is verified and then continuity is checked between the two points. If continuity exists then FINDS asks the operator to identify the point clipped or probed and verifies that the continuity should exist. If the continuity is incorrect then a short is diagnosed and the diagnosis is displayed on the TEC terminal. If the continuity is correct then another point may be tested. This is repeated until incorrect continuity is found or the operator specifies *NEXT or *DONE.

If there is a feedback loop and the operator has not found the fault he may enter *NEXT and the suspected track short portion FINDS operation is repeated for the new mode. *NEXT may be entered as many times as there are nodes on the feedback loop. If there are no remaining nodes FINDS terminates. The operator may also terminate FINDS by entering *DONE.

6.6.3.5 Error Recovery Procedure

The error recovery machanism is available to aid the user in restarting FINDS or repeating selected FINDS measurements. Depressing the ERROR RECOVERY pushbutton causes the message:

ERROR RECOVERY PENDING

to be displayed unless Error Recovery is in progress. FINDS completes the present operation and then displays the message:

ERROR RECOVERY IN PROGRESS

and the teletype bell is rung twice.

The operator now has the choice of entering one of the following:

An IC name --------FINDS performs all measurements previously taken on that IC

An IC.pin name --- FINDS performs all μvoltage measurements including intra-nodal measurements previously taken on that IC. pin.

*REPEAT ---------- FINDS performs the last recoverable measurement made. This is the measurement displayed as last observation unless FINDS is performing a track short test. During track short test the measurement repeated is the last μvoltmeter measurement.

*START ------------Restarts the complete FINDS operation

NOTE

Error recovery only applies to μvoltmeter and intra-nodal measurements. Adjacent track short tests are not included in error recovery.

Error recovery performed, once final diagnosis is reached, is only successful at changing the diagnosis if a new diagnosis is discovered. The most effective use of error recovery at this time is to enter *START to completely re-start the FINDS operation.

6.6.4 Running FINDS

NOTE

When attempting to diagnose a board using the FINDS techniques the operator should be familiar with the FINDS test flow (see para. 6.6).

Before entering FINDS, the operator must ensure that FINDS is enabled (F=ON) and if desired, the other two options pertaining to FINDS, Track Short Detection (T) and Low sensitivity (L) are specified as desired (see para. 6.3.6). When the Guided Probe has produced an intermediate nodal diagnosis display and the operator wants to enter FINDS, depressing the XMIT key on the TEC terminal allows FINDS to be entered and produces a FINDS display (see Figure 6-21).

FINDS directs the operator to place the probes and/or clips via the TEC display.

NOTE

After each positioning of a clip or probe the PROBE pushbutton must be depressed. This indicates to FINDS that the positioning is complete.

When the clips/probes are in position, FINDS makes the necessary measurements and displays another set of directions for the operator. This process is repeated until FINDS determines the fault. At this time FINDS displays the diagnosis and terminates the operation.

While operating FINDS there is one display (see Figure 6-23) that might not be self explanatory. This display directs the operator to probe using PROBE+ on the shoulder of an IC pin and PROBE- on the solder pad connecting that IC pin to the etch board. It is displayed when the system is in the μ-voltage test phase of its strategy. See Figure 6-24 for a complete understanding of this operator instruction.

NOTE

1. Unlike other FINDS operations, the μvoltage measurements are time consuming. It is very important to hold probes in place until a new command or a diagnosis is issued by the system.
2. For this portion of the test, probe positioning is critical. The user must use the plus (+) and minus (-) probes as directed or an invalid diagnosis may result.
3. If at the end of the suspected track short operation FINDS was unable to refine the diagnosis further the probe diagnosis along with any stacked diagnosis from the μ-voltage operation are displayed.

6.7 AFL TESTING

Automatic Fault Location (AFL) features a two step process for automatically determining the source of a board failure without operator intervention. The AFL flow of control is shown in Figure 6-25 and described in the following paragraphs.

```
MSSGE 1:

MSSGE 2:


                          **************************************
                         **                                    **
                        **  PROBE +                ON SHOULDER  **
PROBE:                 **            E43.11                      **
                        **  PROBE -                ON SOLDER PAD **
                         **                                    **
                          **************************************

STATUS:
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

LAST OBSERVATION:  E69
WORKING ON:  E43.11                         CURRENT DIAGNOSIS  HIGH BUT ACTIVE
```

Figure 6-23. Probe Display for u-Voltage Test.

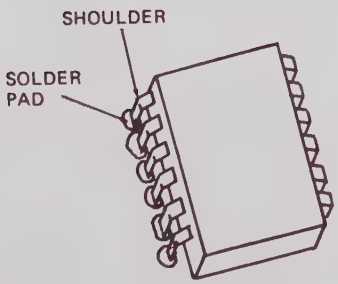


Figure 6-24. IC Probe Placement.

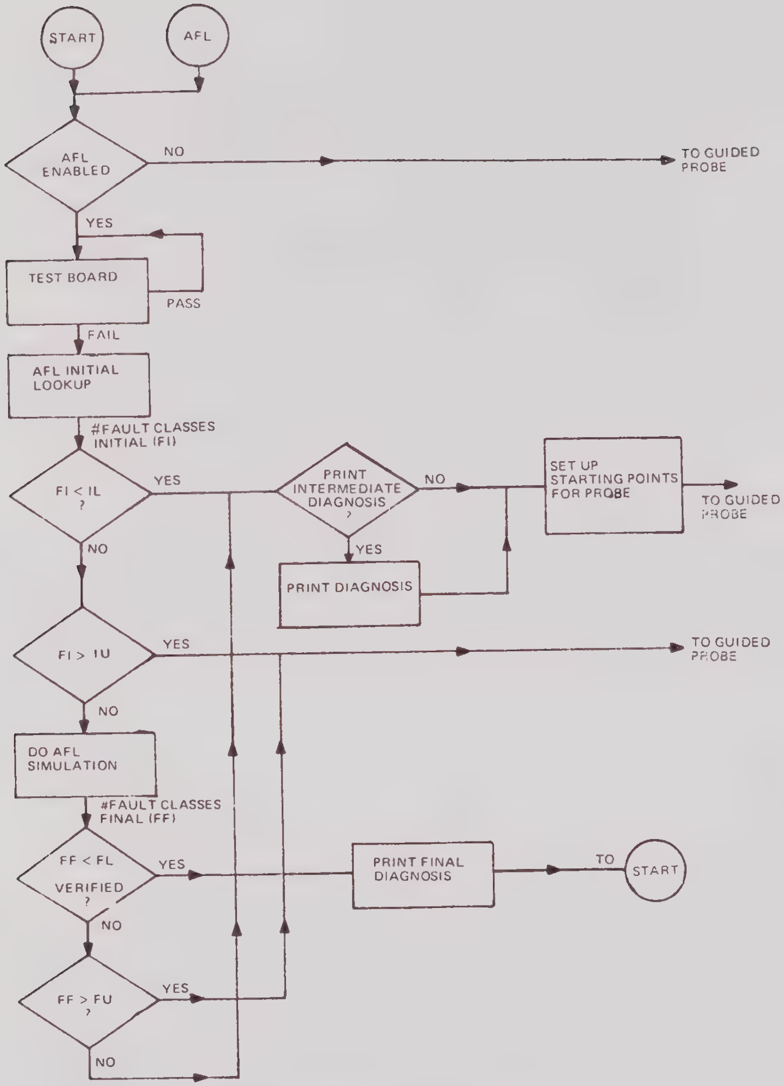

START
AFL
AFL ENABLED
NO
TO GUIDED PROBE
YES
TEST BOARD
PASS
FAIL
AFL INITIAL LOOKUP
#FAULT CLASSES INITIAL (FI)
FI < IL ?
YES
PRINT INTERMEDIATE DIAGNOSIS ?
NO
SET UP STARTING POINTS FOR PROBE
TO GUIDED PROBE
YES
PRINT DIAGNOSIS
NO
FI > IU
YES
TO GUIDED PROBE
NO
DO AFL SIMULATION
#FAULT CLASSES FINAL (FF)
FF < FL VERIFIED ?
YES
PRINT FINAL DIAGNOSIS
TO
START
NO
FF > FU ?
YES
NO

Figure 6-25. AFL Flow of Control.

6.7.1 AFL Options and Flow

Use of the START or AFL buttons cause a transfer to the beginning of the flow chart. If AFL is enabled, then testing can occur. If AFL is not enabled, then AFL is not used and the enabling of the guided probe mode is checked.

Board testing is then performed. If the START button was originally depressed to reach AFL, then testing is started automatically. If the AFL button was used, then the initiation of testing starts only after the START button is depressed. If the UUT passes, then the process loops back and waits for a START button to initiate further testing.

If the UUT fails then the data from the UUT testing is used in an AFL initial lookup. This module looks up those faults which could be responsible for the failure in the fault dictionary. For discussion purposes the number of such faults will be signified by the variable FI.

The AFL INITIAL TRANSFER LIMITS are then used to effect transfers to the probe with or without starting points or to continue to AFL simulation.

If the number of fault classes (FI) is less than the initial lower limit (IL) then a transfer to Intermediate DRM or to the guided probe with starting points is indicated. First, however, the state of the print intermediate diagnosis (I on the PRINT OPTIONS Page) is checked; if it is ON then the faults concerned are displayed. Whether these intermediate faults are printed or not, starting points are prepared for transfer.

If the number of initial fault classes (FI) is greater than the initial upper limit (IU) then a direct transfer to the guided probe is made without preparing starting points.

If neither of these limits causes a transfer to the guided probe, then AFL simulation is performed. This module then classifies the number of faults based on the simulation results. If the number of faults generated by the simulator (FF) is less than the final lower limit (FL) and the faults exactly match the faulty UUT's operation, then a final diagnosis is printed and the overall testing process is restarted.

If the number of fault classes is greater than the final upper limit (FU) then a direct transfer to the guided probe without starting points is made. If both final limit checks fail, the sequence of transferring to the guided probe with starting points is taken.

The PRINT DIAGNOSIS boxes are actually a simplified version of the full diagnosis printing mechanism shown in Figure 6-26.

In Figure 6-26, a single TEC screen is formatted and displayed on the TEC. If the PRINT ON LISTING DEVICE option (D on the PRINT OPTIONS page) is ON then the contents of the TEC screen is copied to the listing device. The listing device is specified in the LI specification of the FILE USAGE page.

The PRINT DIAGNOSES LONGER THAN 24 LINES option (L option on the PRINT OPTIONS page) is then checked. If it is ON and if there are more faults, then the entire process is repeated for the remaining faults. If the L option is OFF or if no faults remain with the L option ON, then the fault printing process is terminated.

6.7.2 Initial AFL Lookup

The initial AFL lookup uses two pieces of data from the UUT testing: the failing test number and the failing vector. That is, the test program step at which a failure first occurs and a list of the driver/sensors which failed at that test.

The fault dictionary prepared in the PREPARE mode is then used. This dictionary holds data showing how each modelled fault should be detected and is organized first by step number, then by failing vector. Thus, at testing time the entries in the fault dictionary for the UUT's failing test are extracted. Only those entries whose failing vectors match the UUT's failing vector are considered possible causes of the UUT's failure.

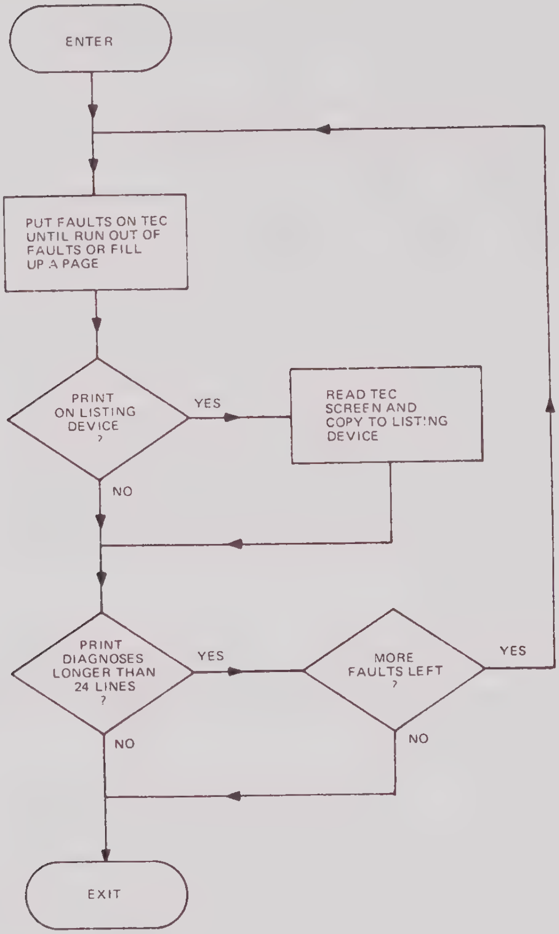

ENTER
PUT FAULTS ON TEC UNTIL RUN OUT OF FAULTS OR FILL UP A PAGE
PRINT ON LISTING DEVICE ?
YES
READ TEC SCREEN AND COPY TO LISTING DEVICE
NO
PRINT DIAGNOSES LONGER THAN 24 LINES ?
YES
MORE FAULTS LEFT ?
YES
NO
NO
EXIT

Figure 6-26. AFL Diagnosis Printing.

In other words, the fault dictionary lookup pulls out those modelled faults which, in simulation, failed at the same step and in the same way as the failing UUT. This process serves to narrow down the number of faults possibly responsible for a particular UUT's failure from hundreds or thousands to about ten. In some cases the resolution at this point might be quite good - that is, the number of possible faults is small due to the fact that different faults manifest themselves in different ways. In other cases, where the UUT fails early in the test program or few outputs from the board are visible, the resolution may be poorer and require AFL simulation to narrow down the number of possible faults.

6.7.3 AFL Simulation

The AFL simulation process is much like the fault simulation performed in the PREPARE mode. That is, faults are inserted in the simulation model of the UUT and the operation of the faulty UUT is then simulated. In the PREPARE mode simulation, the operation of the faulty model's simulation is compared against the simulation of the good (non-faulty) network for the purpose of determining fault detection. In AFL simulation, the faulty model's simulation is compared against the actual faulty UUT's operation. If a faulty model's simulation matches the UUT's operation, then the modelled fault is verified as being responsible for the actual UUT's failure. Any fault whose model's simulation does not exactly match the faulty

UUT's cannot be solely responsible for the UUT's failure. The AFL process, however, inserts only one failure per model (this should not be confused with parallelism during fault simulation) whereas there can exist two, three or more actual failures on a UUT.

It is possible, then, for a modelled fault to be present on a faulty UUT but in some way mis-match the UUT's operation. This is most likely due to one or more additional failures which contribute additional failing tests and pins or which modify the behavior of the first failure. In addition, the mis-match may be due to a non-logical failure; that is, a failure which causes the UUT to behave in a random, unpredictable manner. Examples of this latter category are: a short across an inverter which causes a fast oscillation, or a short resulting in an overload which causes a threshold voltage situation.

AFL simulation utilizes a sophisticated technique which measures the probability that a modelled fault is somehow responsible for a UUT's failure. This technique results in one of three types of diagnosis: verified, verified but others and probable.

"Verified" indicates that the modelled fault exactly matches the UUT's faulty operation.

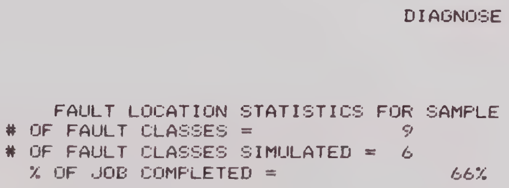
"verified, other faults present" indicates that the modelled fault's failures with respect to a good UUT are a subset of the faulty UUT's failure. This means that the fault(s) indicated in the diagnosis are present on the UUT, and that one or more additional faults are present. These extra faults will be diagnosed only after the fault(s) in the current diagnosis are removed.

"Probable" indicates that the modelled fault's failures were not a sub-set of the UUT's failures but were close enough so as to indicate that the fault is "probably" present.

During the AFL run, the status message shown in Figure 6-27 will occur. The message is updated automatically as the run progresses (after every six fault classes have been simulated) thus reflecting the current status of the run.

6.7.4 The AFL to Probe Link

It is possible to transfer to Intermediate DRM or the Guided Probe after initial lookup or after AFL simulation. If a user wishes to pass starting points resulting from an initial lookup only, and not continue on to the final AFL simulation, it will be faster to turn AFL OFF and use the Quick Lookup feature of the PROBE (rather than using the initial AFL lookup). If a user wishes to have the ability to continue on to a final AFL simulation, and thus narrow down the number of fault classes remaining, then AFL should be run rather than the Quick Lookup feature of the PROBE.



```
                              DIAGNOSE MODE


          FAULT LOCATION STATISTICS FOR SAMPLE
        # OF FAULT CLASSES =             9
        # OF FAULT CLASSES SIMULATED =   6
          % OF JOB COMPLETED =             66%
```

Figure 6-27. DIAGNOSE Mode AFL Status Message.

If AFL is selected, it will be possible to transfer to the PROBE (if enabled) after an initial lookup or after completing the AFL simulation. If the number of fault classes after an initial AFL lookup is small, a transfer to the PROBE with or without starting points can occur.

The AFL to Probe Link performs the task of instructing Intermediate DRM or the Guided Probe as to where to probe initially based on the fault classes under investigation. The Link does this by selecting one node from each fault class. A fault class is a group on one or more individual faults (stuck at zero, stuck at one, short to another pin) which are indistinguishable as observed from the UUT's external behavior. Although the Guided Probe can look at any one of the pins included in a fault class, it must be given a single point to look at, one which will be bad if any of the individual faults in the fault class is present.

Intermediate DRM clips, when instructed, those IC starting points supplied by the Link. If the fault is not found, the starting points are then transferred to the Guided Probe. The Guided Probe, when instructed by the Link, first looks at the points given it. If any of these points are bad, the Guided Probe follows its normal mode of isolating a failure - but starts at the bad point. This is an advantage over the normal Guided Probe start-up procedure which involves tracing back from a failing external output. The use of the Link allows the Guided Probe to jump immediately to the area of failure and often saves many minutes of probing.

It is also possible to utilize the Link after AFL simulation. Even after simulation there might be more than one fault class indicated in the final diagnosis. The Link can then be used to instruct the Guided Probe to determine the exact failure. Indeed, even an AFL final diagnosis involving only one fault class can involve many nodes; the Link can be used to find the exact cause of the failure.

6.7.5 Use of AFL to Probe Limits

The AFL to Probe limits provide a great deal of flexibility in the mixture of AFL and Guided Probe techniques used. The setting of the IL, IU, FL and FU parameters to achieve a minimal diagnosis time is dependent on parameters associated with each UUT. The factors affecting these parameters and advice on their usage is the subject of this paragraph.

Experience with CAPS/APG has shown that the probing of a single point takes, on the average, 10-15 seconds (4-6 probes/minute) and that the minimum probing time for a UUT is 1-1 1/2 minutes. Another fact to keep in mind is that the starting points handed to the Guided Probe by AFL or the quick Lookup feature are "guesses" as to where the failure is located. On the average, fewer than half of these points need be probed before an actual failure is found.

An important consideration in utilizing AFL simulation is that six possible fault classes are simulated in parallel. (This is not true in cases where "by-one" simulation is used for large RAM memory boards; in this case one fault is simulated at a time). Therefore, simulation of one fault class takes as long as six, seven as long as twelve, and so on. The use of AFL simulation usually results in only one or two fault classes diagnosed. The time to probe these fault classes to a final diagnosis must be included in any time estimate using AFL simulation.

Before setting the four AFL parameters, two decisions must be made: whether to use AFL at all vs. Guided Probe and the Quick Lookup feature only, and whether to use AFL simulation vs. AFL initial lookup only.

The first decision is highly dependent on the circuit complexity of the UUT. Highly complex boards can take a great deal of time to probe if starting points are not used. Large circuit depth, feedback loops, wired-and nodes and other circuit configurations can require a great deal of probing in order to "pass through" them to a source of failure, to say nothing of what is required to diagnose failures within these configurations. These types of boards benefit greatly by the use of the AFL to Probe Link or the Quick Lookup feature of the Probe because this allows the Guided Probe to jump to the immediate area of the problem. On the other hand, many boards consist of only a few levels of logic and are not very complex. These types of boards usually do not take very long to probe and any advantage that the Link might have is lost in the overall overhead associated with AFL. In summarizing, the more complex a board is, the more likely AFL will be useful.

The second decision is largely dependent on the time required to perform a set-of-six AFL simulation. In most cases it is faster to let the Guided Probe "look" at the initial lookup fault classes than to let AFL simulation check them. Since it takes only 30-45 seconds to probe three faults (half of six), AFL simulation time must be faster than this for a set of six faults in order for AFL simulation to save time.

Figure 6-28 is a graph which attempts to relate the quantities discussed above. The vertical axis represents diagnosis time and the longitudinal axis is in units of number of fault classes under consideration.

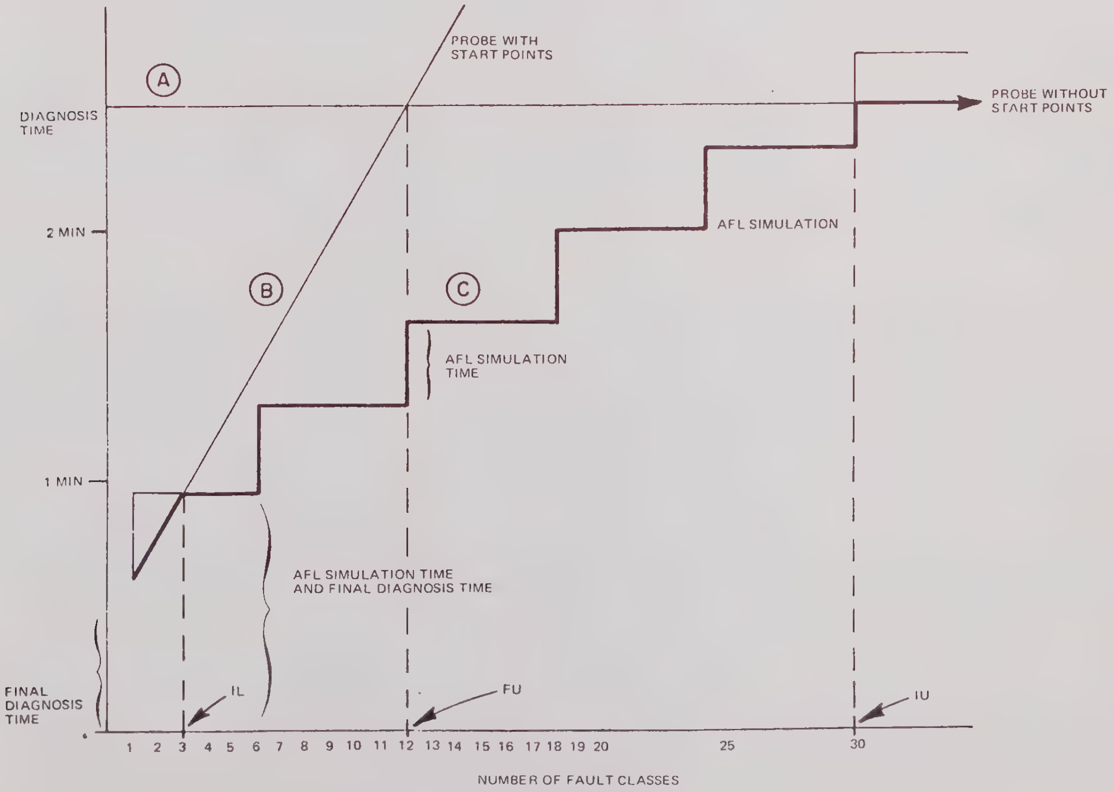


Figure 6-28. Use of AFL Transfer Limits.

The horizontal line Ⓐ is the time that is required to probe without starting points. Line Ⓑ is the time required to probe using starting points. This line slopes at 10-15 seconds per pair of fault classes (on the average only one out of every two must be probed; 10-15 seconds is required per probe). This line is offset from zero by the time that the Guided Probe requires to make a final diagnosis. This quantity is included in all graphs. Step Ⓒ is the time required for AFL simulation and subsequent probing of one or two fault classes. The size of the step is the time required to simulate six faults in parallel. This time can be observed from the data presented on the TEC display while running AFL simulation.

Using this type of graph, the objective is to follow the lowest curve so as to minimize the time required to diagnose. This minimum curve is accented in Figure 6-28.

The IL limit (3 in this example) can be set at the point at which Ⓑ and Ⓒ intersect. Before this point, jumping to the probe after the AFL initial is fastest. The minimum curve then follows the AFL simulation escalator until it hits a line Ⓐ. This intersection is where IU should be set (30 in this example) because more than this number of faults would take longer to simulate than would be required to probe the board without any AFL assistance.

Setting of the FL limit is a matter of user preference. It is intended to be used to suppress linking to the Guided Probe if fewer than some number of faults are verified (exactly matched). This may be useful when visual inspection or other methods will yield faster results than probing these points. Experience has shown that it is always useful to link to the Guided Probe to isolate the failure. This is recommended because AFL's best possible resolution is a fault class which may contain a number of faults; faults which only the Guided Probe can distinguish.

6.8 BOARD TESTING IN MANUAL SUB-MODE.

If the MANUAL sub-mode is enabled by itself, then only this sub-mode can be entered when the control panel START button is depressed (selection of the control panel AUTOMATIC and GUIDED PROBE buttons has no effect). MANUAL sub-mode testing by itself is the technique by which the test system operator uses the controls and displays provided by the system and the test program.

Programs expecially written for use in this sub-mode may direct operator probing and do computations to create their own diagnostic messages. Either the EDIT and TRANSLATE (or AUTOPROGRAM) modes may be utilized to prepare test programs for use in this sub-mode. A translated binary version of the test program must be available prior to running the MANUAL sub-mode. Methods of preparing test programs are described in the 1792-0576 Programming Instructions. Use of the EDIT, TRANSLATE, and AUTOPROGRAM Modes are described in the system operating instructions.

Once the DIAGNOSE Mode options have been selected and the RUN command issued, a READY TO DIAGNOSE message will occur (if the translated binary file is available). At this point, an operator may install the correct device adaptor and/or UUT and depress the control panel START button to initiate testing.

If a failure occurs, the MANUAL sub-mode diagnostic message displays described in para 6.4 can be used to rapidly isolate the failure. In addition, the interactive control panel buttons described in Section 1 can be used to single step or loop, restart on a failure or good condition, etc.

Generate Mode – Section 7

7.1 INTRODUCTION

The CAPS system simplifies the task of entering a circuit description by employing a high-level language and by requiring a user to specify only the inter-IC-connections shown on the schematic diagram. This is made possible through the use of an extensive IC library that contains a full set of integrated-circuit descriptions covering:

1. Their full functional behavior.
2. Internal logic connectivity and feedback paths from one pin to another.
3. Information concerning logic fault models for different logic families.
4. Logic family information with specifications of input and output (high and low) logic voltage threshold values.

Models for over 1900 different IC types (7400, 7474, etc.), covering a wide variety of logic families (TTL, CMOS, Schottky, etc.) are available in two CAPS IC Libraries (DSK:CAPSX.CL and DSK:CAPSXA.CL) furnished with each test system. Models in the IC Libraries are available for immediate use and can be coded directly into the network description (.SR file). Updates, additions, or changes to the CAPS IC Libraries are made by entering elements into an Update Library (DSK:UPLIBX.CL). The CAPS/APG Software will check the User Library (DSK:USERLB.CL) first for the IC named. If the IC cannot be located in the User Library, the Update Library (DSK:UPLIBX.CL) is checked next. If the IC cannot be located in the Update Library, the CAPS IC Libraries (DSK:CAPSX.CL and DSK:CAPSXA.CL) are then checked. The two CAPS IC libraries are referred to in this section as the CAPS IC Library.

The CAPS IC Library also contains a series of special-purpose elements that can be used as basic "building blocks" to construct custom devices or elements not included in the Library. Elements created by a user are placed in a user library (DSK:USERLB.CL). New elements can be added to the user library, deleted, or modified at any time through use of the procedures contained in this section. In addition, IC models included in the CAPS User Library can be coded directly into the network description (.SR file).

The process of IC modeling for the user library is greatly simplified because the user employs the same "network description language" to describe the IC model as was used to describe circuits for testing. For example, a model for a Type 74161 counter exists in the Texas Instruments catalog, consisting of gates and flip-flops. This circuit can immediately be described to the CAPS user library generator by preparing a circuit description (.SR) file as outlined in Section 2. The general principles behind IC modeling are discussed in para. 7.2. A detailed step-by-step procedure and rules for IC modeling are presented in para 7.3. Once the IC is modeled, it needs to be added to the user library. The user must add it to the user library (DSK:USERLB.CL), rather than the system IC library (DSK:CAPSX.CL) so that the integrity of the system IC library can be guaranteed. The user may create several different "user libraries". For example, one could be created for each board, and given the name BOARD.CL, where "BOARD" is the name of the board being tested. Furthermore, user libraries may reside on any disk device. The process of adding IC models to the user library is outlined in para. 7.4. When a circuit description is processed that calls for a modeled IC, a lookup is automatically made first in the user library. If the IC is not found there, a lookup is then made in the system IC libraries. If it is desired to by-pass the user and Update Libraries and always look in the IC libraries for an element, a special flag can be added to the type name field when the element is coded into the network description (.SR) file described in Section 2. For example, adding a + sign to a 7400 (+7400) identifies the element as an element that is to be located only in the IC Libraries.

7.2 IC MODELING -- GENERAL PRINCIPLES

For each Digital IC, there is usually a logic model or diagram of it listed in the manufacturer's catalog. Usually such a model is present in terms of gates and flip-flops. Other models may be in the form of a truth table, or there may be only a verbal description and timing diagrams available. The CAPS System is designed to accept all of these types of descriptions. In particular, if a logic model exists in terms of gates and flips-flops, then the user will describe the model of the IC in terms of such basic elements, called "BOOT" elements. The user will use the "BOOT" elements as if they were ICs and describe the model in the same high level language that is employed when describing circuit descriptions. Since the "BOOT" elements are not standard ICs, the user needs to know their logical behavior and pin configuration. Para. 7.8 provides a detailed description of all "BOOT" elements.

Modeling IC's in terms of "BOOT" elements is the most common way of entering new IC's into the CAPS System. In some cases (decoders, encoders, and ROMs, for example), the user may be faced with a truth table, rather than a logic model. The CAPS System allows a user to enter truth table Type ICs into the library through a process called ROM generation. Truth tables may be entered either in binary, octal, or hexadecimal numbers.

Other more complex elements can be modeled efficiently by a CAPS Special Simulation Package (SSP). SSP code allows each element to be simulated as a complete entity rather than through simulation of the many individual "gates" that might make them up. SSP-BOOT elements such as a shift register, adder, etc. are present in the CAPS IC Library and may be used in the same manner as the simple NAND boot elements described earlier. SSP devices in the library should be used to build other ICs whenever possible as they are more efficient than basic "gate level" devices.

From the discussion, it is clear that the user models the functional behavior, of digital logic. Other data such as LOGIC FAMILY, LOADING and LOGIC TIMING information are also added to the IC elements. Such considerations are further discussed in para. 7.6.

7.3 PREPARING IC MODELS

7.3.1. Basic Techniques

The user must first prepare a circuit description (.SR) file for the IC. The .SR file describes the functional logic of the IC and is very similar to the circuit descriptions prepared for any logic board as discussed in Section 2. Specific examples of network descriptions prepared to insert IC models into the library are presented in the following paragraphs together with some general guidelines to follow when preparing any IC network description. The following general guidelines apply:

1. First, list all pins of the IC to be modeled as EXTernal connections (Inputs, Outputs, V_{CC}, GND, etc.). List all pins (even unconnected pins). As an example, Figure 7-1 lists the dual in-line pin connections for an SN7440 (two 4-Input positive NAND gates). EXT connections for the SN7440 would be as follows:

```
1    EXT      ;
2    EXT      ;
3    EXT      ;
4    EXT      ;
5    EXT      ;
6    EXT      ;
7    EXT      , GND;
8    EXT      ;
9    EXT      ;
10   EXT      ;
11   EXT      ;
12   EXT      ;
13   EXT      ;
14   EXT      , VCC;
```

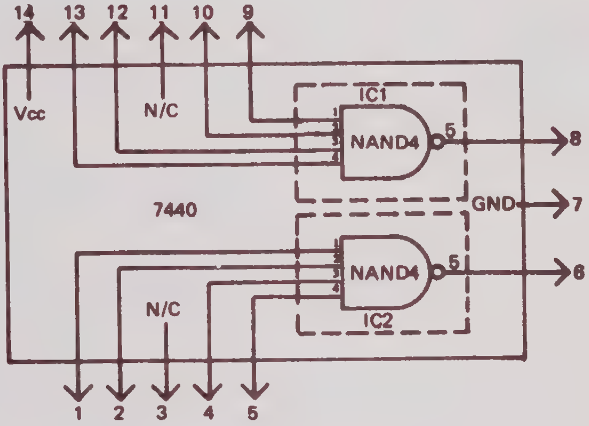


Figure 7-1. Dual In-Line Pin Connections For A 7440.

Note that IC VCC and GND EXTernal connections must be listed as VCC and GND in the CONNECTIONS Field. EXTernal statements for IC pin lists have the following additional restrictions:

a. Normally, an IC cannot have more than 24 Input or Output pins. For ICs which do have more pins, special IC modeling considerations apply as discussed in para 7.6.3.

b. The IC pin number must be inserted in the NAME field. For example, pin number 1 would be listed as:

```
1     EXT     ;
```

2. SECOND, code the IC "BUILDING BLOCKS" or "BOOT" elements described in para. 7.8 into the network description. For example, the SN7440 shown in figure 7-1 could have been coded as follows:

```
IC1     NAND4     9,10,12,13,8;
IC2     NAND4     1,2,4,5,6;
```

3. THIRD, terminate the network description with a dollar sign ($). The terminator can be the first entry in the CONNECTIONS column, and must be followed either by a carriage return or the name information.

4. FOURTH, add the name information as follows:

```
$
NAME=7440;
```

The name information is specified by typing "NAME=" followed by any number of names (up to 8 alphanumeric characters) that are separated by commas and terminated by a semicolon, for example:

```
$
NAME=7440,SN7440N,XYZ;
```

5. Fifth, add the IC type information as follows:

```
TYPE=TTL;
```

The default type TTL will be assumed automatically if it is not specified at this time. IC types may be TTL, CMOS, TTL, Low Power, Schottky etc., A list of all possible IC types and their abbreviations to be used after "TYPE=" includes:

TTL: Transistor-Transistor Logic
LTTL: Low power TTL
HTTL: High power TTL
STTL: Schottky TTL
LSTTL: Low power Schottky TTL
ECL: Emitter Coupled Logic
DTL: Diode-Transistor Logic
RTL: Resistor-Transistor Logic
CMOS: Complementary Metal Oxide Semiconductor
NMOS: N-Channel Metal Oxide Semiconductor
PMOS: P-Channel Metal Oxide Semiconductor

Note that the user must enter "TYPE=" followed by the IC type and terminate the string by either a semi-colon(;) and carriage return or just a carriage return.

6. Sixth, add the PINTYPES information. If desired (for example on a level converter IC) the TYPE of individual pins may be overridden by use of PINTYPES. For example,

```
PINTYPES=ECL, ECL, 8=CMOS;
```

specifies that pins 1 and 2 should be assigned logic type ECL and pin 8 logic type CMOS. The unmentioned pins will be associated with the logic type specified by TYPE.

7. Seventh, add the pin loading data. TTL default loading (inputs are 1 load and Outputs 10 loads) will be assumed automatically if pin loading is not specified at this time. Pin loading for the SN7440 shown in Figure 7-1 might have been specified as follows:

```
LOAD = 1,1,4=1,1,30,8=30,1,1,12=1,1;
```

Where Input pins 1,2,4,5 and 9,10,12,13 are specified as 1 load and Output pins 6 and 8 are specified as 30 loads.

A completed circuit description for the 7440 is shown in Figure 7-2. The SN7440 is logically identical to the SN7420 except for the loading information. For such cases the user may specify both loadings as follows:

```
$
NAME=7440;
TYPE=TTL;
LOAD=1,1,4=1,1,30,8=1,1,12=1,1;
$
NAME=7420;
TYPE=TTL;
LOAD=1,1,4=1,1,10,8=10,1,1,12=1,1;
```

Thus, multiple "dollar sign" sections may appear.

7.3.2 Additional Attributes

Additional attributes can be associated with an IC name by enclosing them in "( )" immediately following the name.

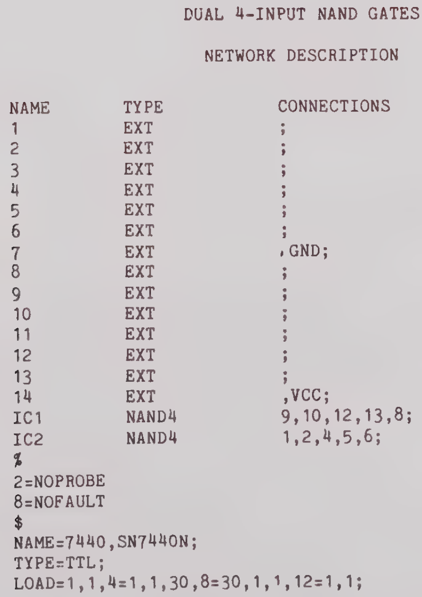
```
$
NAME=7499 (FAKE, TIMING)
```

The attributes that can be specified in this manner are:

```
FAKE
NOPROBE
NOCLIP
NOFUN
NONREC
TIMING
```

The "FAKE" Attribute

The FAKE attribute identifies those ICs that do not exist on the real board but for some reason are added to a network description. ICs identified as FAKE will not be probed or faulted. The attribute is similar to the FAKE command des-



```
                    DUAL 4-INPUT NAND GATES

                      NETWORK DESCRIPTION


NAME          TYPE              CONNECTIONS
1             EXT               ;
2             EXT               ;
3             EXT               ;
4             EXT               ;
5             EXT               ;
6             EXT               ;
7             EXT               ,GND;
8             EXT               ;
9             EXT               ;
10            EXT               ;
11            EXT               ;
12            EXT               ;
13            EXT               ;
14            EXT               ,VCC;
IC1           NAND4             9,10,12,13,8;
IC2           NAND4             1,2,4,5,6;
%
2=NOPROBE
8=NOFAULT
$
NAME=7440,SN7440N;
TYPE=TTL;
LOAD=1,1,4=1,1,30,8=30,1,1,12=1,1;
```

Figure 7-2. 7440 Network Description.

cribed in Para. 2.3.5.4. When the FAKE attribute is specified, it is not necessary to use the FAKE command in the network description. The attribute FAKE can be invoked as follows:

```
$
NAME=DELAY4(FAKE);
```

The "NOPROBE" Attribute

Certain ICs on a board can never be probed. For example, the voltage difference on a germanium diode may be no more than 0.2 V and therefore the probe may not sense this voltage difference. In such cases, the attribute "NOPROBE" should be used as follows:

```
$
NAME=DIODE(NOPROBE);
```

The "NOCLIP" Attribute

If an IC cannot be clipped, then "NOCLIP" must be specified. Note that NOCLIP does not imply NOPROBE.

The "NOFUN" Attribute

Inserting the fault, "IC NOT FUNCTIONING" will be inhibited if the NOFUN attribute is specified:

```
$
NAME=PULLUPS(NOFUN);
```

The "NONREC" and "TIMING" attributes are described in Para. 7.6.

Multiple Attributes

If multiple attributes are needed for a name, the user may add them by separating each one with a comma as follows:

```
$
NAME=RESET(TIMING,FAKE);
```

If multiple names exist, all relevent attributes must be repeated:

```
NAME=RESET(FAKE,TIMING),LOWINIT(FAKE,TIMING);
```

Other Attributes

It is also possible to add some attributes to specific EXTernal pins of the IC when it is being entered into the library. This is done by using the percent processing feature of CAPS. Note, that the percent (%) processing must precede the $NAME.

```
%
2=NOPROBE;
8=NOFAULT;
$
NAME = 7440.
```

would add the attributes NOPROBE to pin 2 and NOFAULT to pin 8 of the IC that is being entered into the library. Adding an attribute to an EXTernal pin causes that attribute to be associated with the corresponding IC pin when it is called out of the library for use. The following percent processing attributes can be added in this manner:

NOFAULT
NOFAULT0
NOFAULT1
NOPRINT
NOPROBE
NOUV

Note, that you can specify up to 13 IC pins per line.

Detailed descriptions of each attribute are listed under percent processing in Para. 2.3.5.

7.3.3 Pin Loading Data

For 54/74 logic, the following pin loading data applies:

a. Each input of a multiple emitter input transistor sinks 1.6 mA at a logical zero and is considered to be one load.

b. Each input sinks 40 µA at a logical one level.

c. The logical one output sources 400 µA.

d. The logical zero output sinks 16 mA.

e. Thus, standard 54/74 output pins are capable of driving current to 10 loads.

f. Reset, set, and clock inputs which normally sink 80 µA are considered to be two loads.

g. The buffer gates 54/7440 fan-out is 30 loads.

All loading information is family normalized. To insert fractional loads, use the format:

nn.mm

Where:

nn may be a one or two digit number from 0 to 63.

mm may be a one or two digit number from 0 to 99.

The decimal point is omitted when there is no fracitonal part. A loading of 63.99 is interpreted as "never overloaded".

7.3.4 A More Advanced Example

The 7474 dual D Type flip-flop may be modeled using the basic "building block" or BOOT" element DPE. Two DPE building blocks are used to construct the 7474 shown in Figure 7-3. The completed circuit description for the 7474 appears in Figure 7-4. In the loading information, only those pins were specified for which the default assumption (1 for inputs, 10 for outputs) did not apply.

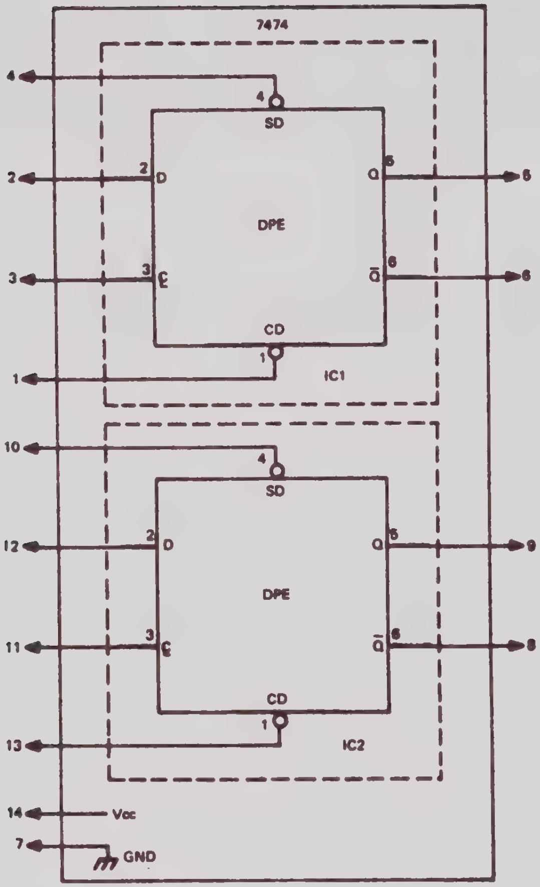

7474
4
4
SD
2
2
D
Q
5
5
DPE
3
3
C
Q̄
6
6
CD
1
1
IC1
10
4
SD
12
2
D
Q
5
9
DPE
11
3
C
Q̄
6
8
CD
13
1
IC2
14
Vcc
7
GND

Figure 7-3. Dual In-Line Pin Connections For A 7474.

```
               DUAL D-TYPE EDGE-TRIGGERED FLIP-FLOPS

                        NETWORK DESCRIPTION


NAME       TYPE            CONNECTIONS
1          EXT             ;
2          EXT             ;
3          EXT             ;
4          EXT             ;
5          EXT             ;
6          EXT             ;
7          EXT             ,GND;
8          EXT             ;
9          EXT             ;
10         EXT             ;
11         EXT             ;
12         EXT             ;
13         EXT             ;
14         EXT             ,VCC;
IC1        DPE             1,2,3,4,5,6;
IC2        DPE             13,12,11,10,9,8;
$
NAME=7474,SN7474N;
TYPE=TTL;
LOAD=1=2,4=2,10=2,13=2;
```

Figure 7-4. 7474 Network Description.

7.4 ADDING IC MODELS TO THE USER OR UPDATE LIBRARIES

7.4.1 Introduction

New IC models (or existing models that are to be modified) may be entered into the CAPS User or update libraries by invoking the "GENERATE" mode of the CAPS/APG Monitor. The CAPS User Library (DSK:USERLB.CL) can contain user generated special purpose elements that are constructed from basic "building blocks" in the CAPS IC Libraries (DSK:CAPSX.CL and DSK:CAPSXA.CL). Users may also create several different user libraries, if desired, by following the techniques described in the following paragraphs. The update library (DSK:UPLIBX.CL) is a single library that can contain new or updated IC models. This library provides a means of updating and modifying the CAPS IC system libraries (DSK:CAPSX.CL and DSK:CAPSXA.CL) which cannot be changed or modified themselves. The CAPS/APG software checks the four libraries in the following manner for any IC named:

1. First, the User Library (DSK:USERLB.CL) is checked for the IC specified.
2. Second, the Update Library (DSK:UPLIBX.CL) is checked.
3. Third, the CAPS system IC libraries (DSK:CAPSXA.CL and DSK:CAPSX.CL) are checked.

It is therefore possible to place changes or additions to the CAPS system IC Libraries (DSK:CAPSX.CL and DSK:CAPSXA.CL) in the Update Library (DSK:UPLIBX.CL) and have the CAPS/APG software automatically use the updated version.

In the GENERATE mode, the .SR file (IC model) is first analyzed for syntax and semantics by the pre-processor (see Figure 7-5). If there are errors in the .SR file, EDIT can be invoked automatically (if the E Option is ON) and errors can be listed on the Teletype (if the P Option is ON). The user corrects the errors and types X and XMIT (From EDIT) to re-analyze the corrected .SR file. When the .SR file is error free, the library generator will invoke several software modules that generate all IC data for simulation, fault modeling, and probing.

The result of this process is a library element which will automatically be added to the user or update library specified by the U= option. If there is insufficient space in the library for the new IC, the library is automatically copied, additional space is added, and the library element is then entered. Note that it is not possible to add or change elements in the CAPS IC System Libraries (DSK:CAPSX.CL) or (DSK:CAPSXA.CL) themselves.

7.4.2 Monitor Controls For The GENERATE Mode.

There are two monitor pages for the library generator mode as shown in Figure 7-6. The user must first specify the .SR file name of the IC (N=). Next, the user may specify the name to be assigned to the IC (T=) in case he has not done so in the .SR file via the NAME=option. Note that the name of the IC may be up to 8 alphanumeric characters in length. It is strongly recommended however, that the user specify the name (or names) of the IC in the .SR file rather than through use of the T=option. In this way, the name is permanently associated with the .SR file and it is less likely that ICs will be added to the library with wrong names.

Next, the user must specify the library to which he wants to add this IC (U=). Note that the user may create many user libraries. For example, the user may have a special library associated with each board or with each different logic family. To create a new library (for example, one called LIB.CL), the user must first copy the user library in the following manner:

```
.COP DSK0:LIB.CL<DSK1:USERLB.CL
```

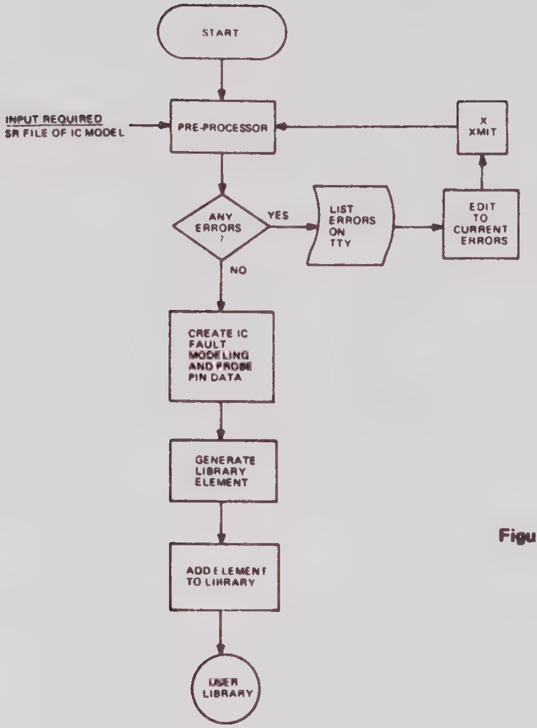

START
INPUT REQUIRED
SR FILE OF IC MODEL
PRE-PROCESSOR
X
XMIT
ANY ERRORS ?
YES
LIST ERRORS ON TTY
EDIT TO CURRENT ERRORS
NO
CREATE IC FAULT MODELING AND PROBE PIN DATA
GENERATE LIBRARY ELEMENT
ADD ELEMENT TO LIBRARY
USER LIBRARY

Figure 7-5. Library Generation.

```
#

        FILE USAGE                                        GENERATE MODE
        ----------

N=      DSK:74XXX.SR    NETWORK FILE
T=      AAAAAAAA        IC ELEMENT NAME (IF NOT IN .SR FILE)
U=      DSK:BDNAME.CL   USER IC LIBRARY FILE
LI=     DSK:            LISTING DEVICE
W=      SYS: , DSK:     WORKING DISKS
X=      SYS: , DSK:     EXTENDED WORKING DISKS
FM=     WIDE            OUTPUT FORMAT (NARROW, WIDE, OR SCOPE)

 TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

```
#

        OPTIONS FOR LIBRARY (COMMANDS: OY, ON, OO, OA)       GENERATE MODE
        --------------------
OPTION  STATUS  DESCRIPTION
  N      ON     LIST NETWORK DESCRIPTION
  W      ON     LIST WARNINGS, UNUSED PINS, AND BOARD STATISTICS
  X      ON     LIST NETWORK CROSS REFERENCE
  P      ON     PRINT (ALL) ERRORS ON TELETYPE
  L      ON     LONG ERROR MESSAGES
  E      ON     INVOKE EDITOR ON ERRORS IN .SR
  C      OFF    CHANGE CURRENT ENTRY WITHOUT ASKING
  R      OFF    ROM GENERATION - ENTER A ROM INTO LIBRARY
  O      OFF    ENTER ELEMENT NON-RECURSIVELY
  B      OFF    ENTER A BOOT ELEMENT IN LIBRARY
  S      OFF    ENTER SSP CODE IN LIBRARY

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 7-6. GENERATE Mode Pages.

Next, the user must specify the listing device, working and extended working disks and output format in a manner similar to that used in the PREPARE mode. On the second monitor page, the user may control print options and edit options. The last three options on this page are unique to library generation and operate as follows:

"C OFF" means that the following message will be printed on the Teletype before an existing element in the library is changed:

```
(NAME) IS ALREADY IN LIBRARY, DO YOU WANT TO DELETE IT?
(TYPE Y OR N).
```

The user must respond by typing "Y" for the IC to be added to the library. If C is ON, a library element will be changed without warning.

The B, R, S and O options should normally be OFF. R must only be ON when ROMs are added to the library (refer to para 7.7). The O options should only be ON if a non-recursive element is to be added to the library and this was not specified in the .SR file (NONREC attribute, refer to para 7.6). Finally, B is used when SCL elements are entered. S is used when special library elements are entered. Again, it is strongly recommended that the user provide this information through the .SR file to avoid any possibility of error and to provide a permanent record of it.

7.5 LIBRARY UTILITY PROGRAMS

7.5.1 Obtaining Library Listings (LIBLIST)

The LIBLIST mode of CAPS/APG is used to obtain a listing of the contents of a library. Listings are available in two forms:

1. A brief form, that lists every device name in the library and the date it was entered.

2. A cross-reference form that includes a list of all names which refer to the same element. With this form the complete list of "equivalent names" is actually printed only once for each element, while all other occurrences of the same name are referenced back (for example, "SEE 7400").

With either form, the output format can be SCOPE, NARROW or WIDE, and when the brief mode is used, several columns are printed. The two forms are shown in Figures 7-7 (brief) and Figure 7-8 (cross-reference).

To use LIBLIST mode, enter it from the top level of the CAPS/APG monitor, set-up the library file to be listed (LB=), the output device (LI=), and the output format (FM=). Next, set the cross-reference option (C=) to NO (N) for brief mode or to YES (Y) for the cross-reference mode. Next, type RUN followed by return (on TTY:) or XMIT (on TEC:). This monitor page is shown in Figure 7-9.

If the LI= option is set to:

```
LI=DSK:
```

the listing will be saved on DSK: under the name LIBLST.LS . The file may be output later through use of the LOOK Mode or any of the other file management CCL commands or Programs (COPY, VERI, PIP, VPIP,etc.).

```
CONTENTS OF CAPSX.CL VERSION NUMBER 900 LAST MODIFIED ON 1/31/79.
----------------------------------------------------------------------
NAME          DATE          NAME          DATE          NAME          DATE
----------------------------------------------------------------------
74159W        1/23/79       7416          1/23/79       7416W         1/23/79
74160         1/23/79       74160W        1/23/79       74161         1/23/79
74161W        1/23/79       74162         1/23/79       74162W        1/23/79
74163         1/23/79       74163W        1/23/79       74164         1/23/79
74164W        1/23/79       74165         1/23/79       74166         1/23/79
74166W        1/23/79       7417          1/23/79       7417W         1/23/79
74170         1/23/79       74172         1/23/79       74174         1/23/79
74174W        1/23/79       74175         1/23/79       74175W        1/23/79
74176         1/23/79       74176W        1/23/79       74177         1/23/79
74177W        1/23/79       74180         1/23/79       74180W        1/23/79
74181         1/23/79       74181W        1/23/79       74182         1/23/79
74182W        1/23/79       74184         1/23/79       74184A        1/23/79
74184AW       1/23/79       74184W        1/23/79       74185         1/23/79
74185A        1/23/79       74185AW       1/23/79       74185W        1/23/79
74190         1/23/79       74190W        1/23/79       74191         1/23/79
74191W        1/23/79       74192         1/23/79       74192W        1/23/79
74193         1/23/79       74193W        1/23/79       74194         1/23/79
74194W        1/23/79       74195         1/23/79       74195W        1/23/79
74196         1/23/79       74196W        1/23/79       74197         1/23/79
74197W        1/23/79       74198         1/23/79       74198W        1/23/79
74199         1/23/79       74199W        1/23/79       7420          1/23/79
7420W         1/23/79       7422          1/23/79       7422W         1/23/79
74221         1/23/79       7423          1/23/79       7423W         1/23/79
7425          1/23/79       7425W         1/23/79       74251         1/23/79
74259         1/23/79       7426          1/23/79       7427          1/23/79
7427W         1/23/79       74279         1/23/79       74279W        1/23/79
7428          1/23/79       7428W         1/23/79       74298         1/23/79
7430          1/23/79       7432          1/23/79       7432W         1/23/79
7433          1/23/79       7433W         1/23/79       74368         1/23/79
74368W        1/23/79       7437          1/23/79       7437W         1/23/79
7438          1/23/79       7438W         1/23/79       7440          1/23/79
7440W         1/23/79       7442          1/23/79       7442A         1/23/79
7442AW        1/23/79       7442W         1/23/79       7445          1/23/79
7445W         1/23/79       7446A         1/23/79       7447          1/23/79
7447A         1/23/79       7450          1/23/79       7451          1/23/79
7453          1/23/79       7454          1/23/79       7460          1/23/79
7470          1/23/79       7472          1/23/79       7473          1/23/79
7473W         1/23/79       7474          1/23/79       7475          1/23/79
7475W         1/23/79       7476          1/23/79       7476W         1/23/79
7480          1/23/79       7483          1/23/79       7483A         1/23/79
7483AW        1/23/79       7483W         1/23/79       7485          1/23/79
7485W         1/23/79       7486          1/23/79       7486W         1/23/79
7489          1/23/79       7489TI        1/23/79       7490          1/23/79
7490A         1/23/79       7490AW        1/23/79       7490W         1/23/79
7491          1/23/79       7491A         1/23/79       7492          1/23/79
```

Figure 7-7. Library Device Names (Brief Form).

The header of the library listing gives the DATE the library was last modified as well as the "library number" for reference. The library number is a number between 1 and 4095, and is incremented each time the library is modified. Should the library be modified when the number is 4095, it will cycle back to 1.

LIBLIST will identify all SSP elements by adding SSP to the end of the element NAME, for example:

"RAMSSP"

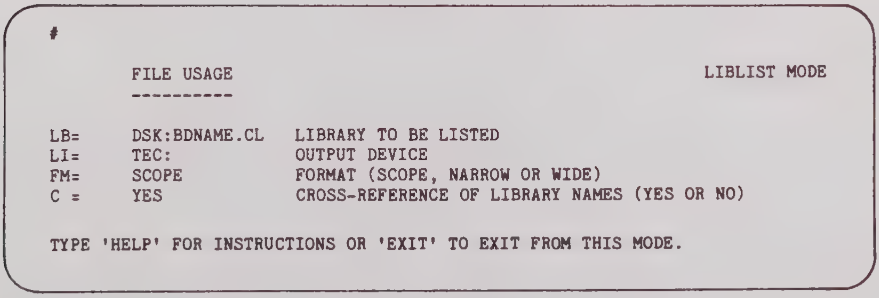
If element NAME has a letter prefix it usually indicates which vendor the element is modeled for. Example:

N 74221 indicates NATIONAL
T 74221 indicates TI
A indicates AMD
S indicates SIGNETICS

No letter prefix indicates compatibility with most vendors.

```
      CONTENTS OF CAPSX.CL VERSION NUMBER 900 LAST MODIFIED ON 1/31/79.
-----------------------------------------------------------------------
NAME        DATE         EQUIVALENT NAMES
-----------------------------------------------------------------------
7404        1/23/79      7405          5404          5405         9016
                         9017          7406          7406W        7416
                         7416W         5406          5406W        5416
                         5416W         7414          7414W        5414
                         5414W         74H04         74H05        54H04
                         54H05         74L04         74LS04       74LS04W
                         74LS05        74LS05W       74LS14       74S04
                         74S04W        74S05         74S05W       54S04
                         54S04W        54S05         54S05W       935
                         936           937           938          9112
                         54L04         54LS04        54LS04W      54LS05
                         54LS05W       54LS14        54LS14W
7404W       1/23/79      7405W         5404W         5405W        74H04W
                         74H05W        54H04W        54H05W       74L04W
                         54L04W
7405        1/23/79      SEE 7404
7405W       1/23/79      SEE 7404W
7406        1/23/79      SEE 7404
7406W       1/23/79      SEE 7404
7407        1/23/79      7417          7417W         7407W        5407
                         5407W         5417          5417W
7407W       1/23/79      SEE 7407
7408        1/23/79      7408W         7409          7409W        3001
                         5408          5408W         5409         5409W
                         74LS08        74LS08W       74LS09       74LS09W
                         906           907           54LS08       54LS08W
                         54LS09        54LS09W
7408W       1/23/79      SEE 7408
7409        1/23/79      SEE 7408
7409W       1/23/79      SEE 7408
7410        1/23/79      7412          7412W         5410         5412
                         5412W         74H10         54H10        74LS10
                         74LS10W       74L10         74S10        74S10W
                         54S10         54S10W        962          963
                         54LS10        54LS10W
7410W       1/23/79      5410W         54H10W        74H10W       74L10W
                         54L10W
74100       1/23/79      74100W
74100W      1/23/79      SEE 74100
74104       1/23/79      74104W        SN29000       GATEDJK
74104W      1/23/79      SEE 74104
74105       1/23/79      SN29001       74105W
74105W      1/23/79      SEE 74105
74107       1/23/79      54107         74LS107       54LS107      54LS107W
74109       1/23/79      54109         54109W        74LS109      74LS109W
                         54LS109       54LS109W
74110       1/23/79      74110W        54110         54110W
74110W      1/23/79      SEE 74110
74111       1/23/79      74111W        54111         54111W
```

Figure 7-8. Library Device Names (Cross-Reference Form).



```
#

        FILE USAGE                                         LIBLIST MODE
        ----------

LB=     DSK:BDNAME.CL   LIBRARY TO BE LISTED
LI=     TEC:            OUTPUT DEVICE
FM=     SCOPE           FORMAT (SCOPE, NARROW OR WIDE)
C =     YES             CROSS-REFERENCE OF LIBRARY NAMES (YES OR NO)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 7-9. LIBLIST FILE USAGE Page.

If the 8000 series elements have a letter suffix, it indicates package type, for example:

```
8288H indicates  A package type
    F indicates  F package type
    W indicates  W package type
```

No letter usually indicates a dual-in-line-package.

7.5.2 Library Element Information (ICINFO)

The ICINFO mode is used to obtain a pin map, connectivity, loading, and size information for a given library element. To use it, enter the major mode ICINFO from the top level of the CAPS/APG monitor, set-up the name of the element to be examined (N=), the library in which it is to be found (LB=), and the listing device (LI=). Then type RUN followed by RETURN (on TTY:) or XMIT (on TEC:). The message "NOT FOUND IN LIBRARY" is the only error response. This monitor page is shown in Figure 7-10.

If the LI= option is set to:

```
LI=DSK:
```

the listing will be saved on DSK: under the name ICINFO.LS. The file may be output later through use of the LOOK Mode or any of the other file management CCL commands or Programs (COPY, VERI, PIP, VPIP, etc.).

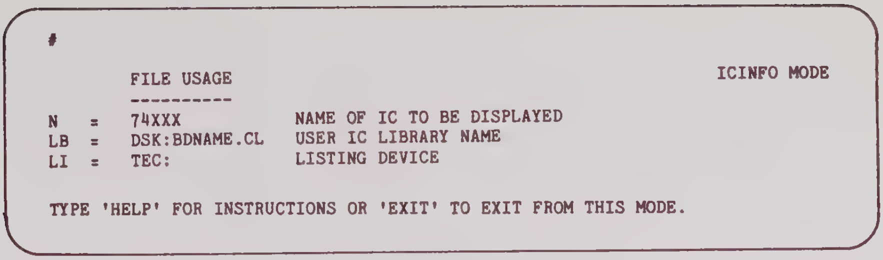
For each pin of the element, ICINFO will report its type (input, output, bus , Vcc, GND, or unconnected), its loading, and its connectivity list. For an input pin, the connectivity list is a list of all the output and bus pins which can be affected by the given input pin. For an output (or bus) pin, the list contains all of the input and bus pins that can affect the given output (or bus) pin. In addition, the ("simtab") size of the element is given in octal notation-sizes for both the recursive and non-recursive versions of the element being given, if relevant. (Note that a recursive element has both a recursive version and a non-recursive version; a non-recursive element has only a non-recursive version). Their sizes can be interpreted as follows: If the element is non-recursive (only one size given), each occurence of that element in a circuit will require (for simulation) the listed number of words in memory. If the element is recursive (both a recursive and a non-recursive size given), each occurence of that element in a circuit will require the recursive size (which is always the smaller of the two) plus an "overhead" of one non-recursive size for the entire circuit. In other words, if there are n occurrences of the element in a given circuit, the total amount of memory required for simulation for these n occurrences is: (n X recursive size) + non-recursive size.

A sample ICINFO output is shown in Figure 7-11.

7.5.3 Library Element Deletion (ICDELETE)

To remove elements from a library, the ICDELETE mode may be used. Elements may not be deleted from CAPSX.CL or CAPSXA.CL but only from a user or update library. There are two forms of deletion:

1. Delete only the specified name. This form removes only the given name from the library directory.
2. Delete the specified name and all equivalent names (that is, all names which refer to the same element) from the library directory.



```
#
        FILE USAGE                                             ICINFO MODE
        ----------
N   =   74XXX            NAME OF IC TO BE DISPLAYED
LB  =   DSK:BDNAME.CL    USER IC LIBRARY NAME
LI  =   TEC:             LISTING DEVICE

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 7-10. ICINFO FILE USAGE Page.

```
      7400          1/10/79                   CREATED FROM: 7400.V8
IC FOUND IN LIBRARY  CAPSX.CL

SAME MODEL USED BY:  7403 7426 9002 7437 7437W 7438 7438W 74132
        74132W 74H00 74H01 74L00 74L03 74LS00 74LS00W 74LS03 74LS03W
        9L00 9L00W 74LS38 74LS38W 74S00 74S00W 74S03 74S03W 74S132
        74S132W 74S37 54S37 74S38 946 949 5400 5403 5426 5437
        5437W 5438 5438W 54132 54132W 54H00 54H01 54L00 54L03
        54LS00 54LS00W 54LS03 54LS03W 54LS37 54LS37W 54LS38 54LS38W
        54LS26 54LS26W 74LS37 74LS37W 74LS26 74LS26W 54S00 54S00W
        54S03 54S03W 54S132 54S132W 9012 9012W 74C00 54C00

PIN      TYPE              LOAD             CONNECTIVITY
-------------------------------------------------------------------------
1       INPUT(TTL)         1.00          3
2       INPUT(TTL)         1.00          3
3       OUTPUT(TTL)       10.00          1, 2
4       INPUT(TTL)         1.00          6
5       INPUT(TTL)         1.00          6
6       OUTPUT(TTL)       10.00          4, 5
7       GND
8       OUTPUT(TTL)       10.00          10, 9
9       INPUT(TTL)         1.00          8
10      INPUT(TTL)         1.00          8
11      OUTPUT(TTL)       10.00          12, 13
12      INPUT(TTL)         1.00          11
13      INPUT(TTL)         1.00          11
14      VCC
```

Note that equivalent ICs using the same model are listed, together with the name of the IC circuit description that the model was created from, IC family type, and SIMTAB size. The following additional data may also be listed, when applicable.

1. Assigned attributes by IC pin in the following format:

```
4      BUSS(NPR,NF1)       10.00       1,13,2
5      INPUT                1.00       8
8      OUTPUT(NPR)         10.00       5,7,9
9      INPUT(NF0)           1.00       6,10
```

Where the attributes NOPROBE (NPR), NOFAULT0 (NF0), and NOFAULT1 (NF1) have been listed.

2. SSP Code Modules used, if any. The format is:

```
USES SSP CODE MODULES: OSHSSP,DFFSSP
```

3. A list of overall attributes used. The format is:

```
ATTRIBUTES: NOPROBE, NOFAULT0, NOFAULT1
```

4. The number of one-shot delay elements in the IC. The format is:

```
2 ONE SHOT DELAY(S) IN IC
```

Figure 7-11. ICINFO Pin Connectivity List.

For example, if an element has been entered with two names (A and B) and form 1 is being used to delete "A" ICDELETE would retain the element under the name B. Using form 2 to delete "A" would cause both A and B to be removed and thus the entire element. Furthermore, the two forms are equivalent whenever an element has only one name. Additionally, form 2 prints Teletype messages of all names that are being deleted (that is, all names for the element).

To delete a library element, enter ICDELETE mode from the top level of the CAPS/APG monitor, set up the name to be deleted (N=), the library in which it is to be found (this may not be CAPSX.CL or CAPSXA.CL), and set the E option "OFF" for form 1 or "ON" for form 2. Then type RUN followed by RETURN (on TTY:) or XMIT (on TEC:). The message "NOT FOUND IN LIBRARY" printed on the Teletype is the only error response. This monitor page is shown in Figure 7-12.

When a library element is deleted, the library is not immediately squished (does not shrink in size) unless the S option is ON. If more than one element is to be deleted the S option should be ON only for the last deletion.

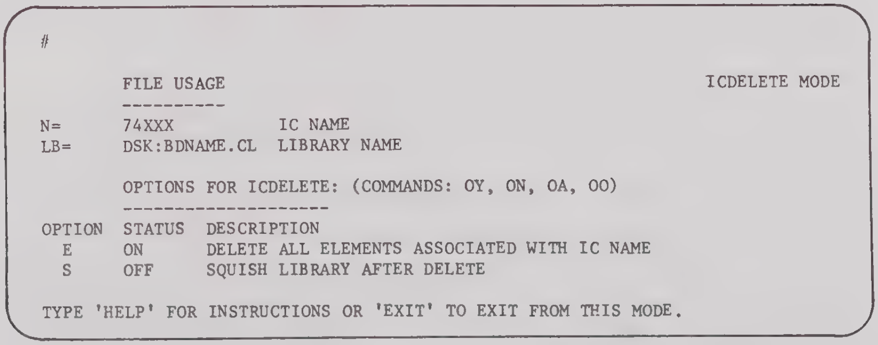
7.6 IC MODELING CONSIDERATIONS

7.6.1 Introduction

The basic technique of modeling IC's was described in para. 7.3. The user was required to generate a network description for the IC, and add name, type and loading information to the end of the file. This was sufficient information for most IC's. The user did not have to know about or consider any other information. In some cases, however, logic timing among IC's is critical, and the user needs to know more about the element he is trying to model. The following paragraphs deal with just such considerations.

7.6.2 Logic Timing Considerations (Recursive Building Blocks)

Generally speaking, each IC can be considered to have a unit delay associated with it. For example, an SN7400 NAND gate's unit delay is approximately 10 n sec in the real world. Similarly, complex MSI elements also have a unit delay between inputs and outputs, independent of their internal model. Sometimes, this unit delay assumption is invalid, and different delays must be associated with the outputs of certain ICs. In these cases, and in cases such as one-shots where the IC's



```
#

        FILE USAGE                                            ICDELETE MODE
        ----------
N=      74XXX           IC NAME
LB=     DSK:BDNAME.CL   LIBRARY NAME

        OPTIONS FOR ICDELETE: (COMMANDS: OY, ON, OA, OO)
        --------------------
OPTION  STATUS  DESCRIPTION
  E     ON      DELETE ALL ELEMENTS ASSOCIATED WITH IC NAME
  S     OFF     SQUISH LIBRARY AFTER DELETE

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure 7-12. ICDELETE FILE USAGE Page.

generate pulses, the user must know how to reflect these timing considerations in his IC modeling.

Let us first introduce the concept of a "building block". A building block is a logic element, which is simulated as an indivisable entity. For example, the NAND gate is a "building block" of the "BOOT" element type. Similarly, the JKPMS flip-flop (JK MASTER-SLAVE) is simulated by executing a dedicated part of the simulator's code. There is no gate expanded model for the flip-flop and therefore it is also a "BOOT" element type building block. The FIFO (as described in para 7.2) is simulated as an entire block. This building block is an example of an "SSP-BOOT" element. A third type of building block is the recursive element. Basically, a recursive element is simulated as if it were a single entity rather than considering its individual elements in detail. It is therefore also a "building block" with a unit gate delay.

It now becomes important to understand the difference between a recursive building block and non-recursive devices. The concept is similar to that of a subroutine for a computer program, where the non-recursive device is the subroutine and the recursive building block is the call to the subroutine. Actually, the non-recursive device is a little network by itself consisting of the original model of the IC with all its gates and flip-flops. It is therefore not a basic building block, but rather consists of many building blocks and thus many delay units.

For most ICs, a recursive building block is also constructed by the library generator at the time that the non-recursive device is generated. The recursive building block consists of input, output and memory records plus a type index. The input, output and memory records contain all logic values of the IC needed for simulation. The type index serves as a pointer to the non-recursive model of the IC. In essence, the recursive element contains all of the logic information relevant at the IC boundary (logic values, memory values, and network connections). The non-recursive element contains all of the information that is internal to the IC.

Thus, for each recursive building block there must always be a non-recursive device. It is the non-recursive device which is actually simulated.using the logic values of the individual copy of the recursive device. An example of how these two models of an IC are used for simulation purposes appears in Figure 7-13. Basically, a simulation table of a network must be constructed. If for example, IC14, IC39 and IC47 are all SN74161's, then the following simulation table would be constructed in core:

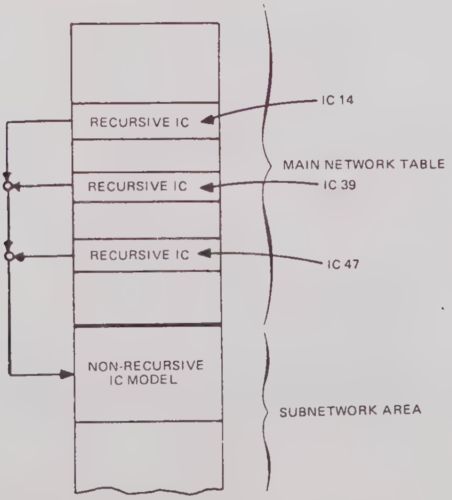


Figure 7-13. Recursive and Non-Recursive Models.

In this simulation table there are three recursive building blocks containing the current logic values of the three ICs. All three recursive elements point to the same non-recursive model of the IC, which resides in an area totally separate from that of the main network table. When simulation is started, the logic values of the recursive IC are used by the non-recursive IC to simulate the proper logic function. The new logic values derived from this simulation will then be used to update the existing logic values in the recursive element. In this manner, all of the internal logic delays of an IC are properly simulated. However, they will have no effect on the main network simulation because only the final values of the IC simulation are used in the main network simulation and thus only one unit delay is apparent.

In summary, the recursive model contains only input, output and memory values and is part of the main network, while the non-recursive model acts like a subroutine which allows the IC to be simulated.

With this understanding, several important concepts relating to logic timing can now be listed:

1. Each "building block" in CAPS has a unit logic delay associated with it. A "building block" may be a basic "boot" element, an "SSP-boot" element, a recursive element or a truth table (ROM) element.

2. If building blocks are chained together, the overall logic delay will be the sum of their delays.

3. An IC will automatically be entered as a recursive building block if it has three or more internal outputs that do not connect to external output pins as shown in Figure 7-14.

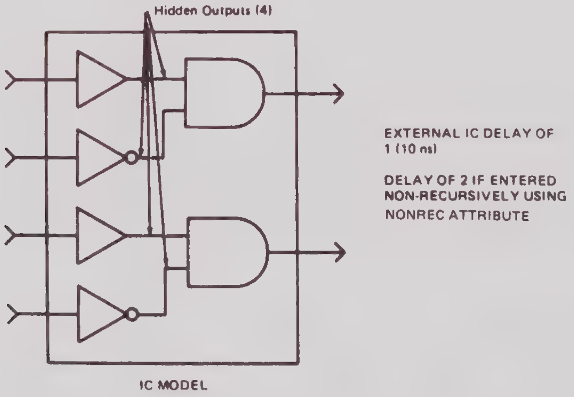

Hidden Outputs (4)
EXTERNAL IC DELAY OF 1 (10 ns)
DELAY OF 2 IF ENTERED NON-RECURSIVELY USING NONREC ATTRIBUTE
IC MODEL

Figure 7-14. Entering Elements Recursively.

4. While a building block will appear to have an external delay (input to output) of one, the internal delays will all be properly simulated because the building block is simulated as an independent network.

5. Because a stimuli to an IC is simulated until the internal IC logic has become stable, pulses do not propagate external to an IC but are propagated internally. A building block that is to be entered into the library is shown in Figure 7-15.

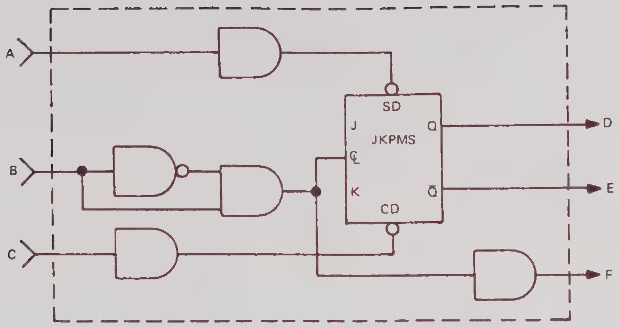


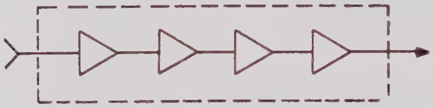
Figure 7-15. Entering A Building Block Into The Library.

If the building block were to undergo a low to high transition at "B" while "A" and "C" were also high, then the flip-flop would toggle. However, "F" will always be low, because the internal pulse at the clock will not be propagated out of the IC, but rather will be given a chance to stabilize.

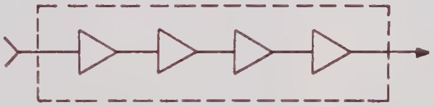
6. To adjust timing, it is always possible to enter an IC as a non-recursive rather than as a recursive building block. This is done by adding the attribute "TIMING" to the name of the IC when generating it:

```
$
NAME=7400(TIMING);
```

In this case, the expanded IC model will be inserted into the main network rather than the recursive building block. A very graphic example serves to illustrate this discussion:



$
NAME=DELAY4(TIMING);

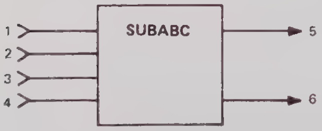


$
NAME=DELAY1;

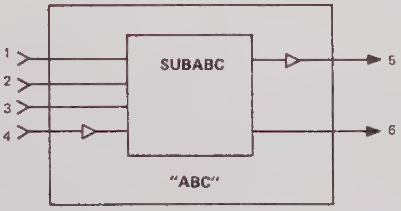
The first circuit will be inserted as a chain of four building blocks in the library with a cummulative delay of 4. The second circuit, however, will have only a delay of one.

7. IC's that generate pulses which must proprogate out of the ICs must be added to the library as a TIMING element.

8. Variable delays are best achieved by first inserting an IC as a recursive block and then creating a second .SR file for the IC, where delays have been added to selected pins of the recursive building block as in the following examples:



$
NAME=SUBABC;



$
NAME=ABC(TIMING);

SUBABC.SR

ABC.SR

In particular, the building block "SUBABC" will always have a unit gate delay independent of its internal model. The second .SR file for the IC consists of the "SUBABC" building block (delay of 1) and two buffers. Because it was inserted into the library with the attribute TIMING, the following timing relationships hold:

	DELAY AT OUTPUT	
CHANGE INPUT	5	6
1	2	1
2	2	1
3	2	1
4	3	2

9. Pulse generating IC's using oneshots must always be inserted as timing elements. Furthermore, any IC using this new element as a building block will automatically have the attribute "TIMING" associated with it. The library generator will always force an IC to have the "TIMING" attribute if it uses any building block which itself was entered as having "TIMING". To allow the user not to carry the attribute "TIMING" forward to other IC's, a second attribute "NONREC" exists. It is identical to "TIMING" except that it does not require that other IC's using the "NONREC" elements themselves become "NONREC".

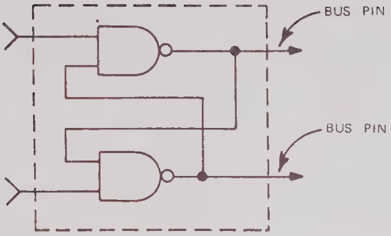
10. The library utility program ICINFO always indicates the core size of both the recursive and non-recursive building blocks. Clearly, the non-recursive building block should be much larger than the recursive ones. Thus, if a recursive IC is used 10 times, the total core requirement will be 10 times the size of the recursive building block plus 1 times the size of the non-recursive building block. This size is usually orders of magnitude smaller than 10 times the size of the non-recursive building block. For those times when memory space needs to be optimized, it may be preferable to use the non-recursive model for an IC that is only used once in a network. The user may specify this manner of optimized memory allocation by adding a star sign (*) before the IC type in the network description. The star sign indicates to the system that it must select the non-recursive model. Note, however, that such a selection, while saving memory, will affect the logic timing as previously noted.

7.6.3 Considerations Relating to Large Pin Counts

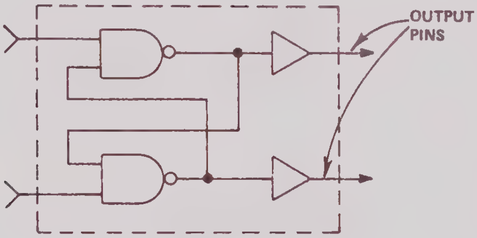
A restriction on the number of IC pins has been set at 24 input pins and 24 output pins for an IC. This restriction exists only for a building block, (that is for a boot element, an SSP boot, a ROM element, or a recursive element). A non-recursive element (TIMING) consists of several building blocks or NONREC and thus does not have this restriction. Thus, ICs having more than 24 input pins and/or 24 output pins must be inserted as a NONREC element, in which case, the maximum number of IC pins will be 63, while the maximum number of input pins or output pins is also 63.

7.6.4 IC Bus Pins

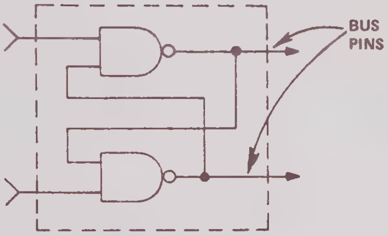
Many ICs have pins which are both inputs and outputs. This is very typical of microprocessor chips, where 4 or 8 pin bus information is received and transmitted. In the CAPS system, the definition of an IC buss pin is broader than normal to allow IC pins to be bus pins even though that is not the normal way to operate the chip. In particular, an IC pin is a bus pin, if the internal logic of the IC drives both the IC pin and another element within the IC as shown in the following example:



Each gate is driving another gate and fans out to an IC pin. Thus, the two output pins of the IC could also be driven externally (backward driving) and the pins are therefore listed as bus pins. Such bus pin situations frequently arise in flip-flops such as the SN7474 or SN7476 and in MSI logic such as shift registers and counters. Both for accurate diagnosis and for fault simulation, the user must decide through his IC model whether a pin is a buss pin or not. He does this by placing a buffer element at the fan-out point for IC output pins that should not be buss pins as shown in the following examples:



BUFFERED IC PINS



UNBUFFERED IC PINS

Since CAPS/APG represents each BUS pin as an input and an output, a library element with 16 pins (two of which are bus pins) will be entered into the library with 18 pins. These additional pins do not count toward the limit of 63 total pins for library elements.

7.7 ROM GENERATION

7.7.1 Introduction

The ROM generator gives the user the ability to create his own unique ROM's via a simple mechanism implemented through a highly structured input file. The ROM created may have from 2 to 4096 addressable locations (the "size" of the ROM), and each location may contain up to 24 bits of information (the "width" of the ROM). Furthermore, addresses may be specified to the ROM generator in either octal or decimal, and the location contents may be specified in binary, octal or hexadecimal. This format flexibility thus frees the user from the drudgery of converting to an arbitrary number system base.

7.7.2 Three Steps of ROM Generation

There are three steps required to completely model a ROM as an IC element for addition to the library. The first step is preparing the input file for the ROM generator; the second step is running the ROM generator; and the third step is using the generated ROM as a "boot" element or "building block" to model a specific IC using conventional library generation techniques.

7.7.2.1 Preparing The ROM Input File (.RG)

A simple, yet flexible input file specifies the name of the ROM element being created, its dimensions (size and width), and its contents. The file is, by convention, given the .RG extension.

The first three "statements" of the .RG file are used to specify the name being given to the ROM, the size of the ROM, and the width of the ROM. Their format is:

```
NAME = "ROMNAME";
SIZE = "ROMSIZE";
WIDTH = "ROMWIDTH";
```

Where:

"ROMNAME" is an alpha-numeric name (of up to 8 characters) that will be assigned to the ROM element as its library name.
"ROMSIZE" is a decimal number between 2 and 4096 inclusive.
"ROMWIDTH" is a decimal number between 1 and 24 inclusive.

An optional fourth element is "RECURSIVE;". ROMGEN normally creates the ROM BOOT with the "timing" attribute turned on. If, however, RECURSIVE is in the .RG file, then ROMGEN will leave the 'timing" attribute off.

The next two statements that may appear in the input file are both optional. They may be used to specify the number base in which addresses will be written (octal or decimal), and the base in which the ROM contents will be written (binary, octal or hexadecimal).

Their format is:

```
ADDRESS BASE = O or D;
CONTENTS BASE = B or O or H;
```

Where B signifies binary, O signifies octal, D signifies decimal, and H signifies hexadecimal.

If no address base specification appears, octal is assumed; while if no contents base specification appears, binary is assumed.

The remainder of the .RG file is the actual specification of the contents for the individual locations of the ROM. Additionally, there is the provision for including a DEFAULT statement which may appear as often as desired. This statement specifies the default contents to be inserted in all locations which are not specified. The format is:

```
DEFAULT = "CONTENTS";
```

Where "CONTENTS" is the default contents to be supplied in unspecified addresses. (Of course, the "CONTENTS" must be written in the contents base.) If no DEFAULT statement has appeared, a default contents of 0 (all zeros) is assumed.

All addresses (which must be specified in increasing order) have the following format:

```
"ADDRESS": "CONTENTS";
```

Where "ADDRESS" is the ROM address (written in the address base) and "CONTENTS" is the contents to be supplied for that address (written in the contents base).

Any addresses that are skipped in the (increasing) sequence of address statements will be filled by whatever the default contents are at the time, that is as specified by the last DEFAULT statement (or 0 if there is no preceeding DEFAULT statement).

The following conditions are file format requirements which must necessarily be understood so as to be able to enter the ROM correctly. The name, size and width parameters must be the first three statements in an .RG file. Furthermore, if any errors are detected in these, or if they are not all present, an error message will occur and all further processing of the file will be inhibited. Next, all statements must end with a ";". Additional statements can be included on the same line, if desired, by incrementing the address by one and separating the entries by a semicolon (;). The address is incremented with the plus sign (+). If the address is incremented by more than one then the address must be specified. See example two. For convenience, any "=" statement need only contain the first letter of the attribute (N= is equivalent to "NAME="). Comments may appear anywhere in the .RG file with a "/" beginning the comment and the end of the line terminating it. Finally, since the size of a ROM must be a power of two, any size not already a power of two is <u>increased</u> to the <u>nearest</u> power of two. The standard statement format is:

```
ADDRESS:CONTENTS; ADDRESS:CONTENTS;/COMMENT
                      or
ADDRESS:CONTENTS; +:CONTENTS;/COMMENT
```

The following examples illustrate the formats used to prepare a ROM.RG File. Line numbers are listed for reference purposes only, and are keyed to the explanations that follow each example.

EXAMPLE 1

```
                                                              Line #

N=EXAMPLE;              /NAME IS EXAMPLE                        1)
S=16;                                                           2)
W= 4;                   /THIS IS A 16 BY 4 ROM                  3)
A=D;                    /ADDRESSES WILL BE SPECIFIED IN DECIMAL 4)
C=0;                    /CONTENTS WILL BE SPECIFIED IN OCTAL    5)
D=17;                   /ALL ONES WILL BE THE DEFAULT FOR NOW   6)
0:11;                                                           7)
1:5;                                                            8)
3:2;                                                            9)
6:7;                                                           10)
D=0;                    /CHANGE THE DEFAULT TO ALL ZEROS       11)
8:12;                                                          12)
9:6;                                                           13)
10:3;                                                          14)
13:4;                                                          15)
14:7;                                                          16)
```

Explanation of ROM.RG: EXAMPLE 1

Line 1 specifies that the name of this ROM is EXAMPLE.

Lines 2 & 3 indicate that it is a 16 by 4 ROM.

Line 4 indicates that all addresses will be written in decimal (legal addresses being 0,1,...,8,9,10,11,...,14,15).

Line 5 indicates that the contents will be written in octal -- the legal contents for a 4-bit wide ROM being 0,1,2,...7,10,11,...,17 in octal.

Line 6 indicates that until another default statement is encountered, any skipped address will contain all ones (i.e., 17 octal).

Line 7 fills location 0 with the bit pattern 1001 (=11 octal).

Line 8 fills location 1 with the bit pattern 0101 (=5 octal).

Line 9 fills location 3 with 0010 and since location 2 has been skipped, fill it with 1111.

Line 10 fills location 6 with 0111 and since locations 4 and 5 have been skipped, fills them with 1111.

Line 11 changes the default to all zeros for all <u>future</u> skipped locations.

Line 12 fills location 8 with 1010 and since location 7 was skipped, fills it with 0000 (all zeros - the new default contents).

Lines 13 & 14 fill locations 9 & 10 with specified contents.

Line 15 fills location 12 with 0000 (since it was skipped) amd fills location 13 with 0100.

Line 16 fills location 14 with 0111.

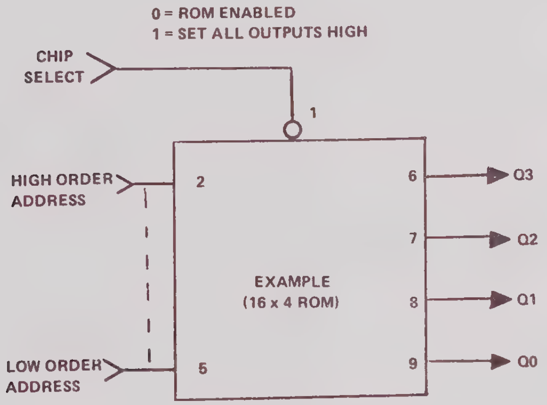
Since the end of the file is encountered next, the remaining locations (only location 15 in this case) are filled with the current default contents of 0000 (see Figure 7-16).

<u>EXAMPLE 2</u>

The same example written with more than one entry per line.

```
                                                        LINE NUMBER
N=EXAMPLE                /NAME IS EXAMPLE                    1)

S=16;                                                        2)

W=4;                     /THIS IS A 16 BY 4 ROM              3)

A=D;                     /ADDRESS WILL BE SPECIFIED IN DECIMAL   4)

C=O;                     /CONTENTS WILL BE SPECIFIED IN OCTAL    5)

D=17;                    /ALL ONES WILL BE DEFAULT FOR NOW   6)

0:11;+:5;3:2;6:7;D=0/CHANGE THE DEFAULT TO ALL ZEROS         7)

8:12;+:6;+:3;13:4;+:7;                                       8)
```

	Contents				
Location	Q3	Q2	Q1	Q0	(Outputs)
0	1	0	0	1	
1	0	1	0	1	
2	1	1	1	1	
3	0	0	1	0	
4	1	1	1	1	
5	1	1	1	1	
6	0	1	1	1	
7	0	0	0	0	
8	1	0	1	0	
9	0	1	1	0	
10	0	0	1	1	
11	0	0	0	0	
12	0	0	0	0	
13	0	1	0	0	
14	0	1	1	1	
15	0	0	0	0	


0 = ROM ENABLED
1 = SET ALL OUTPUTS HIGH
CHIP SELECT
1
HIGH ORDER ADDRESS
2
LOW ORDER ADDRESS
5
EXAMPLE (16 x 4 ROM)
6
7
8
9
Q3
Q2
Q1
Q0

	Q3	Q2	Q1	Q0
6_8 =	0	1	1	1
10_8 =	1	0	1	0

Figure 7-16. The 16 x 4 ROM Example.

Explanation of ROM.RG: example 2

Line 1 through line 6 are exactly the same as in the first example.

Line 7 is the same as lines 7 through 11 in the first example.

Line 8 is the same as lines 12 through 14 in the first example

7.7.2.2 Running The ROM Generator

To use the ROM generator, it is only necessary to turn on the R option on the second monitor page of the major mode GENERATE. The file name, output device and library, as well as the working disks, must also be appropriately set-up on the first page. Furthermore, if the N option (list network description) on the second page is ON, the .RG input file will be displayed on the output device, with any errors detected by the ROM generator prominently displayed. In particular, the message

********** ERROR X ON ABOVE LINE **********

will be displayed under the line on which ERROR number X has been detected. X is a number that ranges from 1 to 9 and corresponds to the following nine errors:

ERROR 1: The address base specification ("A=") is a letter other than an O (octal) or D (decimal).

ERROR 2: The contents base specification ("C=") is a letter other than a B (binary), an O (octal) or an H (hexadecimal).

ERROR 3: A character other than a digit (0-9) was encountered where a numeric character is expected.

ERROR 4: The address base is octal, but an address was specified containing the digit 8 or 9.

ERROR 5: An address larger than 4095 (decimal) was specified.

ERROR 6: An address larger than the size of the ROM was specified.

ERROR 7: An address was specified out of order (i.e., not in increasing order).

ERROR 8: A contents specification has an illegal character or an illegal digit in the contents base.

ERROR 9: A contents specification has more than 24 characters.

7.7.2.3 Using The ROM

The new ROM, when inserted into the library by the ROM generator, is given the following pin configuration:

Pin 1: Chip select input - 0 enables ROM, 1 sets all outputs high (to 1).

Pins 2 to K: Address input lines (pin 2 is the high order, pin K is the low order).

Pins K+1 to N: Output lines (the number of output lines=the width of the ROM).

The number of address lines is clearly dependent on the size of the ROM, as the following table illustrates:

SIZE OF ROM	# ADDRESS LINES
2	1
3 to 4	2
5 to 8	3
9 to 16	4
17 to 32	5
33 to 64	6
65 to 128	7
129 to 256	8
257 to 512	9
513 to 1024	10
1025 to 2048	11
2049 to 4096	12

Once the ROM has been generated, it may then be used as a "boot" element or "building block" to model the specific IC needed via conventional library generation techniques (refer to para 7.3).

7.7.2.4 Automatic ROM/PROM Learning Software (Automatic .RG Generation)

The ROM/PROM Learning Software takes the output data from a ROM, using Driver/Sensors, and builds a .RG file for use in the CAPS GENERATE Mode. The operator writes a simple test program to enable and exercise all locations of the ROM sequentially. The first test step must be at Address 0, and all Addresses must be accessed sequentially. An EXEC Statement within the test program is used to transfer essential information such as NAME, SIZE, and WIDTH of the ROM. After Translation, the test program is run in the MANUAL DIAGNOSE Mode with Miscellaneous Option 'B' (Binary Data Logging) turned on. This program can be run on any GenRad Test System with CAPS X.

A sample test program follows:

```
        BEGIN CAPS 'ABC123';        /BINARY LOGGING IS ONLY LOGGED/
                                    /WITHIN A CAPS RECORDED SECTION/
        SET PS-1 V=5 I=1;           /ANY REQUIRED PS STATEMENTS ARE OK/
        ASS LGC VILA=0 VIHA=3.5
                VOLA=1.5 VOHA=1.5
                IA(5-11) OA(1-4);   /DATA LINES MUST START WITH D-S#1/
                                    /AS MSB AND OTHER DATA OUTPUTS IN/
                                    /ASCENDING DRIVER-SENSOR ORDER TO/
                                    / LSB/
EXEC 'SETUP' ('NAME',64,4);         /THIS OVERLAY BUILDS SETUP.XX ON SYS:/

                                    /NAME = FINAL.RG FILE NAME/
                                    /64 = SIZE OF ROM/
                                    /4 = WIDTH OF ROM/
        1:IL (#)IH(11);             /TEST # 1 IS ADDRESS 0 AND 11/
                                    /HIGH TO ENABLE ROM/
        .SEQ 63(5,6,7,8,9,10);      /SEQUENCE THROUGH ALL LOCATIONS/
FINISH CAPS;
END;
```

1. SETUP.SV and REDUCE.SV must be resident on SYS:.

2. NAME.RG (ROM data file) will be built onto DSK: after REDUCE.SV has run.

3. Outputs must be connected to sensors starting at #1 and be in ascending order.

4. Any initialization must be done with SET LGC statements at Test # 1 or in a FINISH RECORD - BEGIN RECORD section.

5. The first CAPS Test # must be address 0 of the ROM.

6. Each location of the ROM must be addressed starting at address 1 at test step 2.

7. The .RG file can now be entered into your User Library using the GENERATE Mode.

8. It is advisable to use pullup resistors on data outputs.

7.8 BOOT ELEMENTS IN THE IC LIBRARY

A listing of all "boot" elements in the CAPS IC Library can be obtained through use of the Utility Program LIBLIST described in Para. 7.5.1. Each of the major "boot" element types are listed in Table 7-1. The table also references those pages that contain additional information (schematic diagrams, etc.) covering each of the elements.

All names associated with each element are included in the IC library listing. For example, the ground element can be identified as either GND or GROUND in network descriptions and therefore is listed under both names in the library. A cross-reference listing of library names that includes both the assigned name and any equivalent names that apply may be obtained through LIBLIST if the C Option is set to YES.

The library listing also includes a set of software subroutines that are not library elements, but are used by the CAPS software to save memory requirements, adjust logic timing, or otherwise aid in the simulation of library elements, when needed. The following names fall into this category:

1. All names listed as SSn, Sn, or SUBn (such as SS116, S161, or SUB7495).
2. FIFA, FIFF, FIFO, and FIFX names (such as FIFA64X4, FIFF64X4, FIFO32X8, and FIFX64X4).
3. RM184 and RM185 names.
4. LCHND, LTCHR, LTCHX, and X172 names.
5. RSLCH, SLTCH, SLTCHE, JKBAR, LCHND, LTCHR, LTCHX, CADDR, and 1TPMS names.

Table 7-1

TYPICAL BOOT ELEMENTS
(IC BUILDING BLOCKS)

BASIC "GATE LEVEL" ELEMENTS

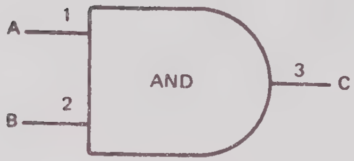
Element	NAME	Page Ref.
AND	Two Input AND Gate	7-30
AND3	Three Input AND Gate	7-30
ANDn	AND Gate with n Inputs	7-30
ANDOR	AND/OR Gate	7-30
AOINV	AND/OR Invert Gate with two Inputs per gate	7-30
AOINV3	AND/OR Invert Gate with three Inputs per gate	7-30
BLOB	Simulates Solder BLOB between two adjacent tracks	7-39
BUF	Buffer	7-33
BUF0	Zero Gate Delay Buffer	7-33
DEL2	Two Gate Delay	7-35
DELn	Delay with n Gate Delays	7-35
DIODE	Diode	7-33
ELTCH	Latch With Enable	7-34
ECLBUS	ECL to TTL Converter	7-62
FIFO	First In First Out	7-60
FILTER	Filter	7-37
GND	Simulates GND	7-39
HFILTER	High Filter	7-37
HLPN	High Low Transition and Pulse Negative Output	7-36
HLPNL	High Low Transition and Pulse Negative Long Output	7-36
HLPP	High Low Transition and Pulse Positive Output	7-36
LATCH	Latch Without Enable	7-34
LFILTER	Low Filter	7-37

Table 7-1 (Con't)

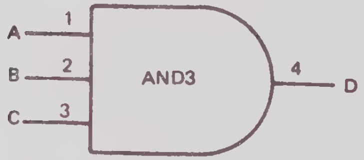
Element	NAME	Page Ref.
LHPN	Low High Transition and Pulse Negative Output	7-36
LHPP	Low High Transition and Pulse Positive Output	7-36
LRC	Line Receiver	7-63
LTCH	Latch Without Enable	7-34
LTCHE	Latch With Enable	7-34
NAND	Two Input NAND Gate	7-31
NAND3	Three Input NAND Gate	7-31
NAND4	Four Input NAND Gate	7-31
NANDn	NAND Gate with n Inputs	7-31
NOR	Two Input NOR Gate	7-32
NOR3	Three Input NOR Gate	7-32
NORn	NOR Gate with n Inputs	7-32
NOT	Inverter	7-33
NOT0	Zero Gate Delay Inverter	7-33
NPN	NPN Transistor	7-33
ONESHT	One-Shot	7-35
OR	Two Input OR Gate	7-32
OR3	Three Input OR Gate	7-32
ORn	OR Gate with n Inputs	7-32
PNP	PNP Transistor	7-33
PULLUP	Simulates PULLUP Resistor	7-39
RESET	Simulates RESET Pulse when SYSTEM POWER is applied	7-39
RESETL	Simulates Long Reset when SYSTEM POWER is applied	7-39
RESNET	Simulates a network of 14 pullup resistors	7-40
RNET	Simulates a network of 6 pullup resistors	7-40
RPAC14	Simulates a network of 13 pullup resistors	7-40
RPAC16	Simulates a network of 15 pullup resistors	7-40
RPAC8	Simulates a network of 7 pullup resistors	7-40
RSNET	Simulates a network of 12 pullup resistors	7-40
RSNT14	Simulates a network of 13 pullup resistors	7-40
RSNT16	Simulates a network of 15 pullup resistors	7-40
RSNT8	Simulates a network of 7 pullup resistors	7-40
SOCKET	Simulates a UUT test socket	-40
TRNSVR	Transceiver	7-63
VCC	Simulates Vcc	7-39
WIRE	WIRE Element (simulated plated-through holes)	7-30
XDET	X Detector	7-39
XGEN	Simulates the X state	7-39
XNOR	Exclusive NOR Gate	7-31
XOR	Exclusive OR Gate	7-31
	SSP IC BUILDING BLOCKS	
ADDRn	n Bit Binary Adder	7-49
CNT4B	Four Bit Binary or Decade Counter	7-57
DECn	n Bit Binary Decoder	7-50
DPE	D Type Positive Edge Triggered Flip-Slop	7-46
DPEP	D Type Positive Edge Toggle with Parallel Input	7-54
DPMS	D Type Positive Master-Slave Flip-Flop	7-45
ENCn	n Bit Binary Encoder	7-51
FSFn	Fast Shift Register (n=100-999)	7-53
FSFTn	Fast Shift Register (n=1-99)	7-53
FSn	Fast Shift Register (n=1000-4000)	7-53
JKPDL	J-K Flip-Flop with Positive Data Lockout	7-43
JKPE	J-K Positive Edge Triggered Flip-Flop	7-42
JKPMS	J-K Positive Master-Slave Flip-Flop	7-41
RAM	Random-Access Memory (Read/Write Memory)	7-56
ROM	Read only Memory	7-21
RSPMS	R-S Positive Master-Slave Flip-Flop	7-44
SFTnA or S	Shift Register with n Bit Asynchronous or Synchronous Load	7-52
TPE	Toggle Positive Edge Triggered Flip-Flop	7-48
TPEP	D Type Toggle Positive Edge Parallel	7-55
TPMS	Toggle Positive Master-Slave Flip-Flop	7-47

AND & AND/OR INVERT GATES

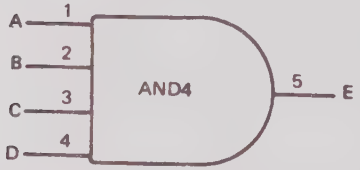
AND
ANDn (n inputs)
ANDOR
AOINV3



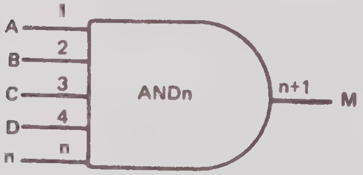
$C=A \bullet B$


A
1
B
2
C
3
AND3
4
D

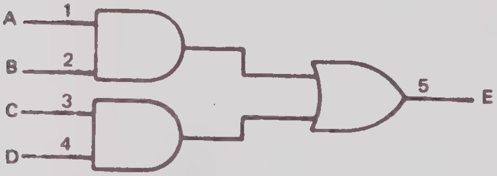
$D=A \bullet B \bullet C$


A
1
B
2
C
3
D
4
AND4
5
E

$E=A \bullet B \bullet C \bullet D$

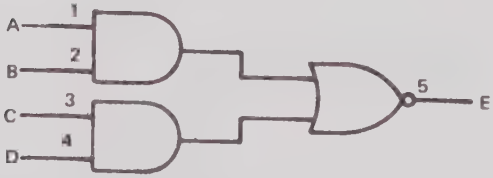

A
1
B
2
C
3
D
4
n
n
ANDn
n+1
M

$M=A \bullet B \bullet C \bullet D \ldots n$


A
1
B
2
C
3
D
4
5
E

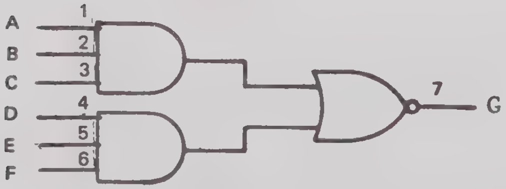
ANDOR

$E=AB+CD$


A
1
B
2
C
3
D
4
5
E

AOINV

$E=\overline{AB+CD}$

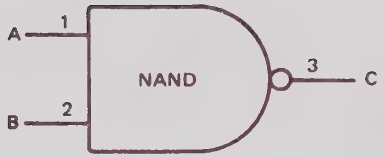

A
1
B
2
C
3
D
4
E
5
F
6
7
G

AOINV3

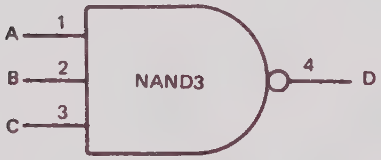
$G=\overline{ABC+DEF}$

NAND & EXCLUSIVE OR GATES

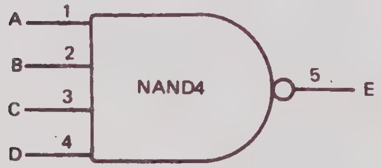
NAND
NANDn (n Inputs)
XOR
XNOR


A
1
B
2
NAND
3
C

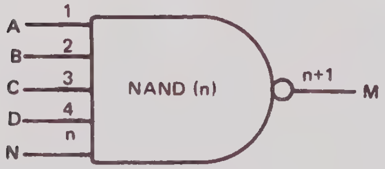
$C=\overline{A \bullet B}$


A
1
B
2
C
3
NAND3
4
D

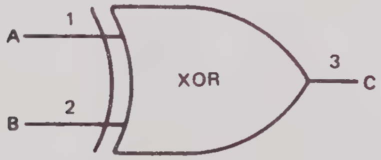
$D=\overline{A \bullet B \bullet C}$


A
1
B
2
C
3
D
4
NAND4
5
E

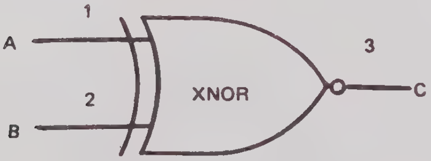
$E=\overline{A \bullet B \bullet C \bullet D}$


A
1
B
2
C
3
D
4
N
n
NAND (n)
n+1
M

$M=\overline{A \bullet B \bullet C \bullet D \ldots n}$



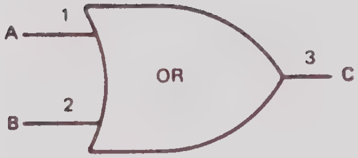
$C=A \bullet \overline{B}+B \bullet \overline{A}$


A
1
B
2
XNOR
3
C

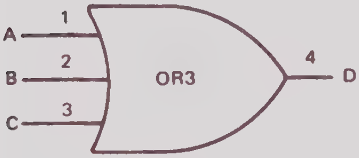
$C=A \bullet B+\overline{A} \bullet \overline{B}$

OR & NOR GATES

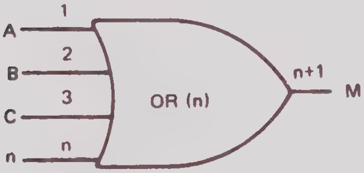
OR
ORn (n Inputs)
NOR
NORn (n Inputs)



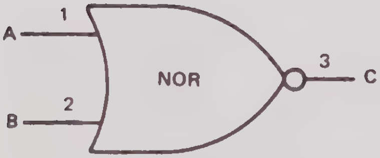
$C=A+B$


A
B
C
1
2
3
OR3
4
D

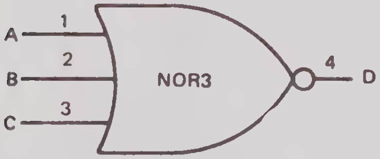
$D=A+B+C$


A
B
C
n
1
2
3
n
OR (n)
n+1
M

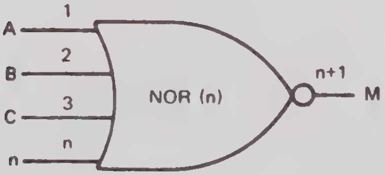
$M=A+B+C+\ldots n$



$C=\overline{A + B}$


A
B
C
1
2
3
NOR3
4
D

$D=\overline{A+B+C}$

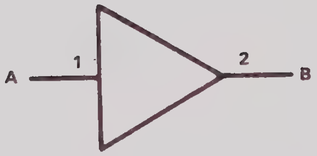

A
B
C
n
1
2
3
n
NOR (n)
n+1
M

$M=\overline{A+B+C+\ldots n}$

BUFFERS, INVERTERS, DIODES, TRANSISTORS

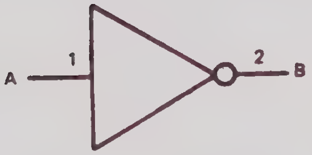
BUF0
BUF
NOT0
NOT

DIODE
NPN
PNP



BUF
BUF0

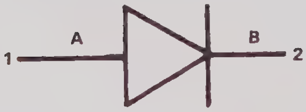
B=A



NOT
NOT0

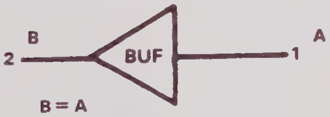
B = $\overline{A}$

BUF0 AND NOT0 ARE ZERO GATE DELAY DEVICES

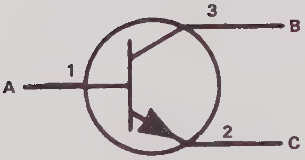


DIODE

Modeled as:

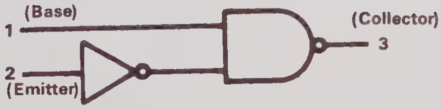


B = A

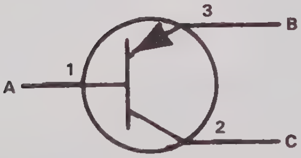


NPN

Modeled as:

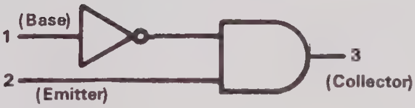

(Base)
1
2
(Emitter)
(Collector)
3

B = $\overline{\overline{C} \cdot A}$



PNP

Modeled as:

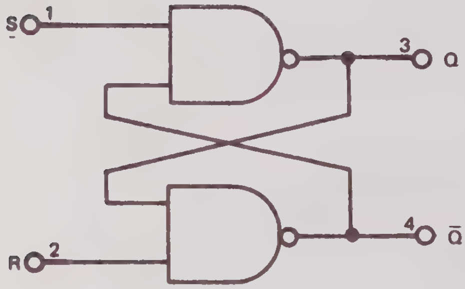

(Base)
1
2
(Emitter)
3
(Collector)

B = $\overline{A} \cdot C$

LATCHES

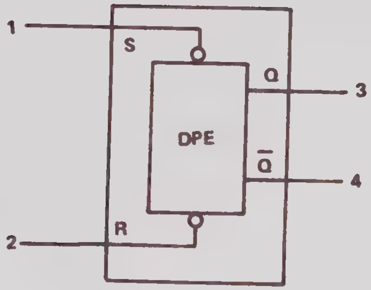
LATCH
LTCHE

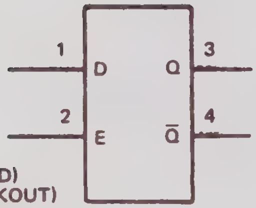
LTCH
ELTCH



LATCH

LTCH

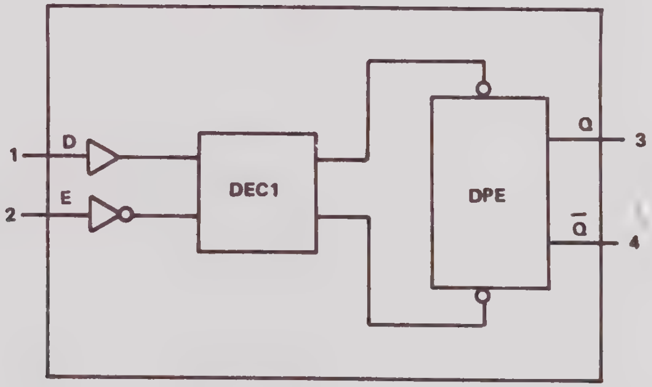

1
S
Q
3
DPE
Q̄
4
R
2



LTCHE

ENABLE
E=1 (LOAD)
E=0 (LOCKOUT)

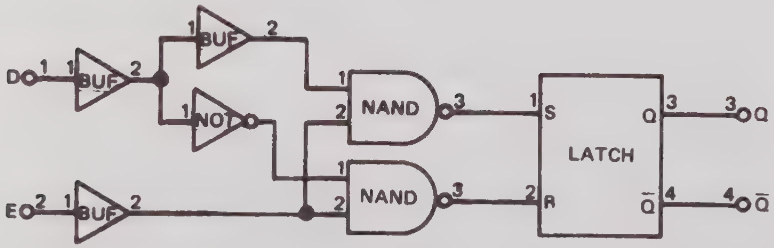
ELTCH


1
D
2
E
DEC1
DPE
Q
3
Q̄
4

ENABLE
E=1 (LOAD)
E=0 (LOCKOUT)

LTCHE

Modeled as:

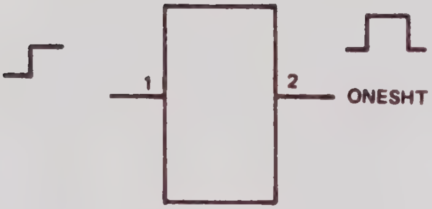

D
1
1
BUF
2
1
BUF
2
1
NOT
E
2
1
BUF
2
1
NAND
3
2
1
NAND
3
2
1
S
2
R
LATCH
Q
3
3
Q
Q̄
4
4
Q̄

LTCHE

ONE-SHOT & DELAY ELEMENTS

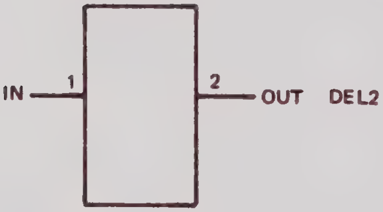
ONESHT

DELn (n gate delays)

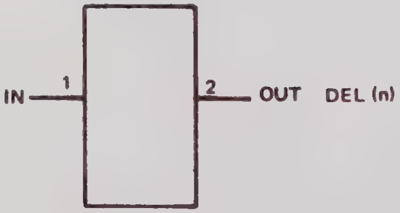


Retriggerable

Time delays for one-shots are specified in terms of typical gate delays (10ns). A maximum of 4095 unit delays are possible (refer to Para. 2.3.7.3.)

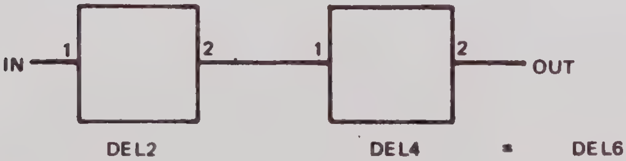

IN
1
2
OUT
DEL2

Two-gate delay. Input shifted by two unit delays.


IN
1
2
OUT
DEL (n)

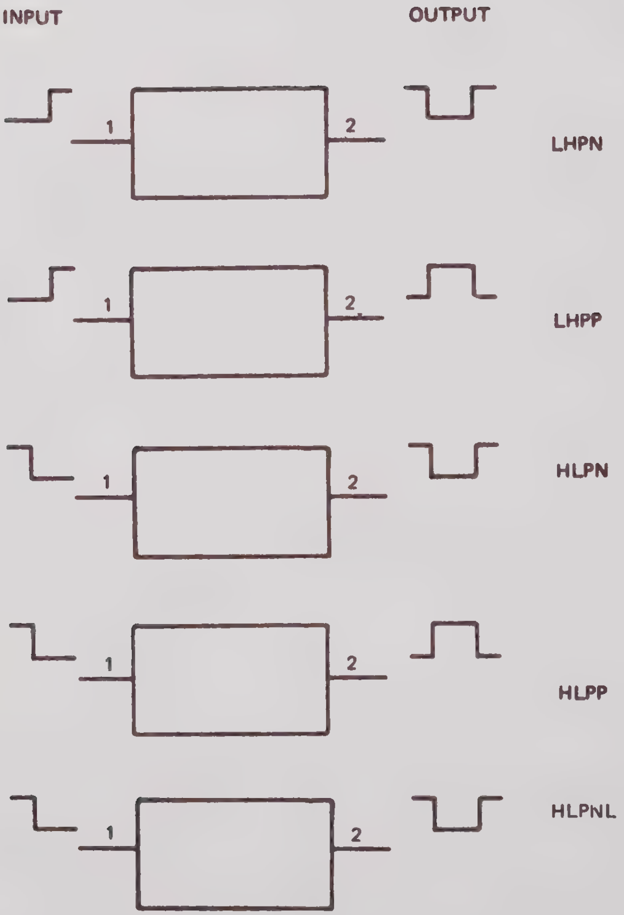
n Gate delay. Input shifted by n unit delays.

LIBRARY DEL ELEMENTS CAN BE CASCADED TO BUILD OTHER DELAYS


IN
1
2
1
2
OUT
DEL2
DEL4
=
DEL6

PULSE ELEMENTS

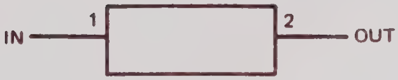
LHPN
LHPP
HLPN
HLPP
HLPNL


INPUT
OUTPUT
1
2
LHPN
1
2
LHPP
1
2
HLPN
1
2
HLPP
1
2
HLPNL

All Output Pulses are 4 delays wide with 2 unit delays between the input edge and first output edge.

Output pulses are 15 delays wide.

FILTER( ) HFILTER( ) LFILTER( )

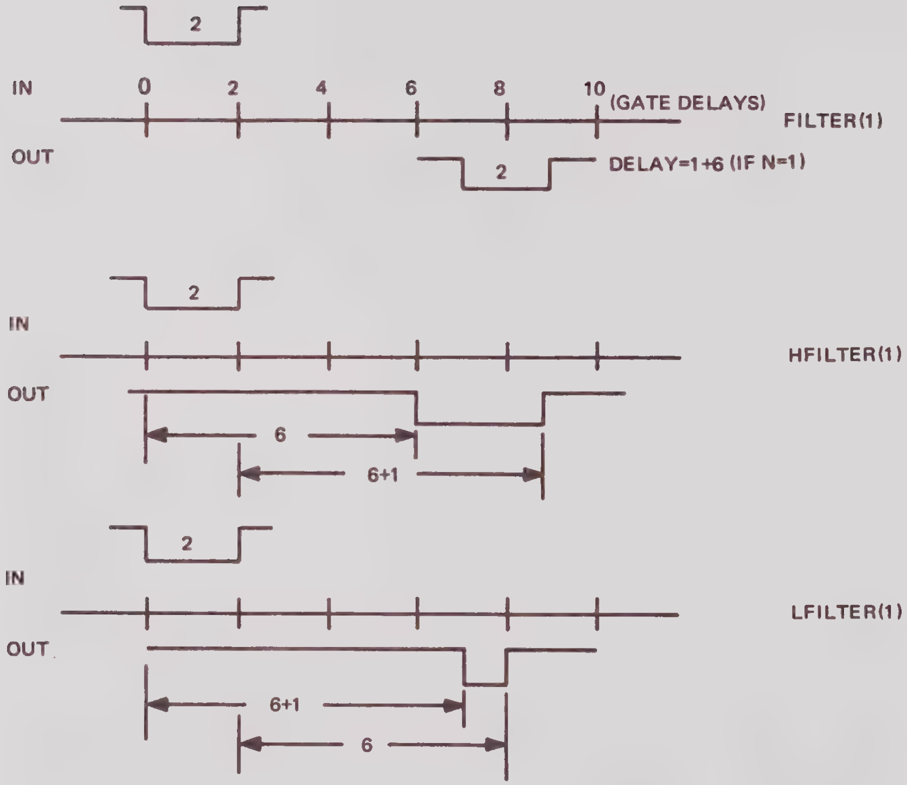


TYPE FILTER	IF INPUT IS A NEGATIVE PULSE	IF INPUT IS A POSITIVE PULSE
FILTER(N)	PASSES > N BED N+6	PASSES > N BED N+6
FILTER(NT)	PASSES > NT BED NT	PASSES > NT BED NT
HFILTER(N)	PASSES ALL TED N+6 LED 6	PASSES > N LED N+6 TED 6
HFILTER(NT)	PASSES ALL TED NT	PASSES > NT LED NT
LFILTER(N)	PASSES > N LED N+6 TED 6	PASSES ALL TED N+6 LED 6
LFILTER(NT)	PASSES > NT LED NT	PASSES ALL TED NT

TERMS USED

N	The Filter Delay Characteristic specified in terms of typical gate delays (10ns for TTL).
NT	The Filter Delay across Test step boundaries.
BED	Both Edges Delayed.
TED	Trailing Edge Delayed.
LED	Leading Edge Delayed.
FILTER	Delays both positive and negative going transitions by N+6.
LFILTER	Delays negative going transitions by N+6 and positive going transitions by 6.
HFILTER	Delays positive going transitions by N+6 and negative going transitions by 6.

TYPICAL FILTER TIME DELAYS

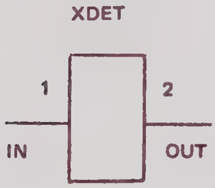

2
IN
0
2
4
6
8
10
(GATE DELAYS)
FILTER(1)
OUT
2
DELAY=1+6 (IF N=1)
2
IN
HFILTER(1)
OUT
6
6+1
2
IN
LFILTER(1)
OUT
6+1
6

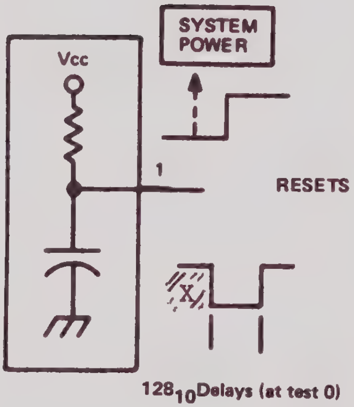
SPECIAL PURPOSE DEVICES

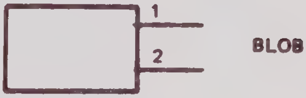
- BLOB
- RESETS
- XDET
- RESETL
- PULLUP
- VCC
- GND
- XGEN
- PULLDOWN

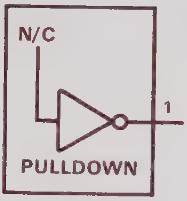
XDET TRUTH TABLE

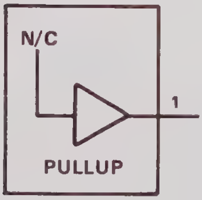
INPUT	OUTPUT
0	0
1	0
X	1

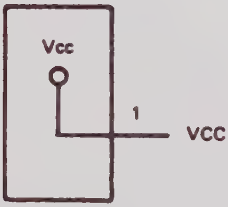

XDET
1
2
IN
OUT

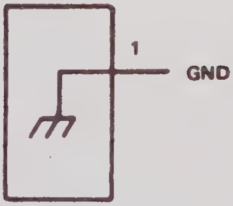

SYSTEM POWER
Vcc
1
RESETS
X
128_{10} Delays (at test 0)

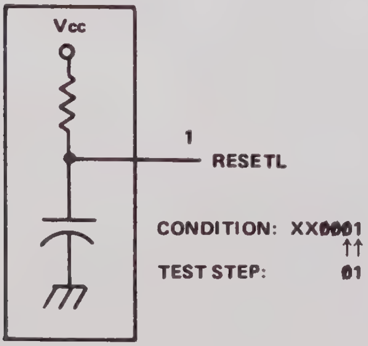


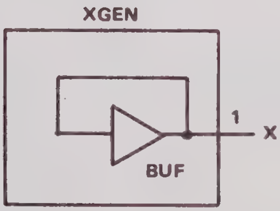

N/C
1
PULLDOWN







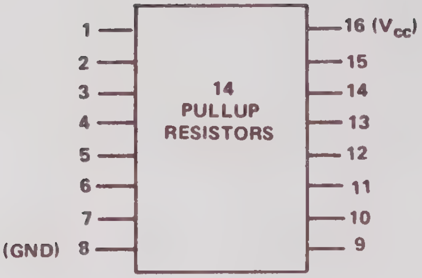

Vcc
1
RESETL
CONDITION: XX0001
↑↑
TEST STEP: 01



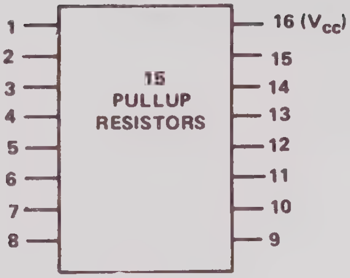
RESISTOR PACKAGES

16 PIN

RSNET

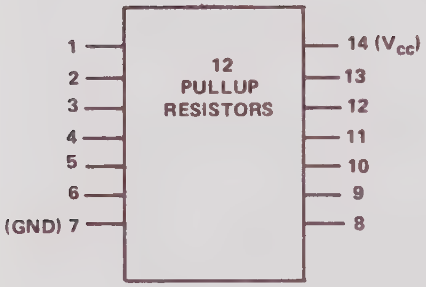

14
PULLUP
RESISTORS
1
2
3
4
5
6
7
(GND) 8
16 (V_cc)
15
14
13
12
11
10
9

RSNT16
RPAC16

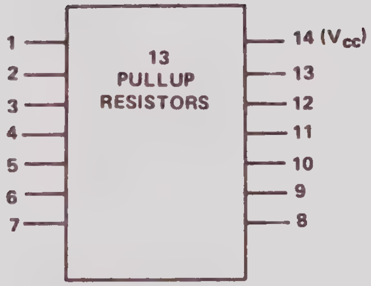

15
PULLUP
RESISTORS
1
2
3
4
5
6
7
8
16 (V_cc)
15
14
13
12
11
10
9

14 PIN

RESNET

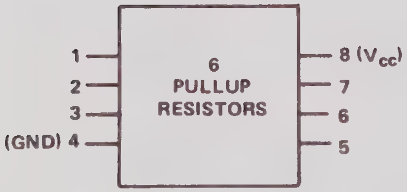

12
PULLUP
RESISTORS
1
2
3
4
5
6
(GND) 7
14 (V_cc)
13
12
11
10
9
8

RSNT14
RPAC14

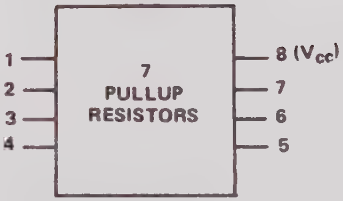

13
PULLUP
RESISTORS
1
2
3
4
5
6
7
14 (V_cc)
13
12
11
10
9
8

8 PIN

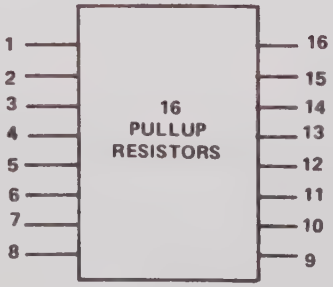
RNET


6
PULLUP
RESISTORS
1
2
3
(GND) 4
8 (V_cc)
7
6
5

RSNT8
RPAC8


7
PULLUP
RESISTORS
1
2
3
4
8 (V_cc)
7
6
5

SOCKET


16
PULLUP
RESISTORS
1
2
3
4
5
6
7
8
16
15
14
13
12
11
10
9

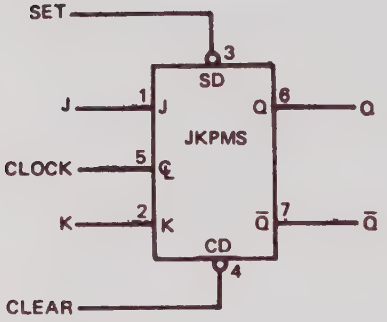
Note, you must connect Vcc and GND in the .SR.

USED WHEN CODING TEST SOCKET ON UUT

J-K POSITIVE MASTER-SLAVE FLIP-FLOP

JKPMS

PIN ASSIGNMENTS

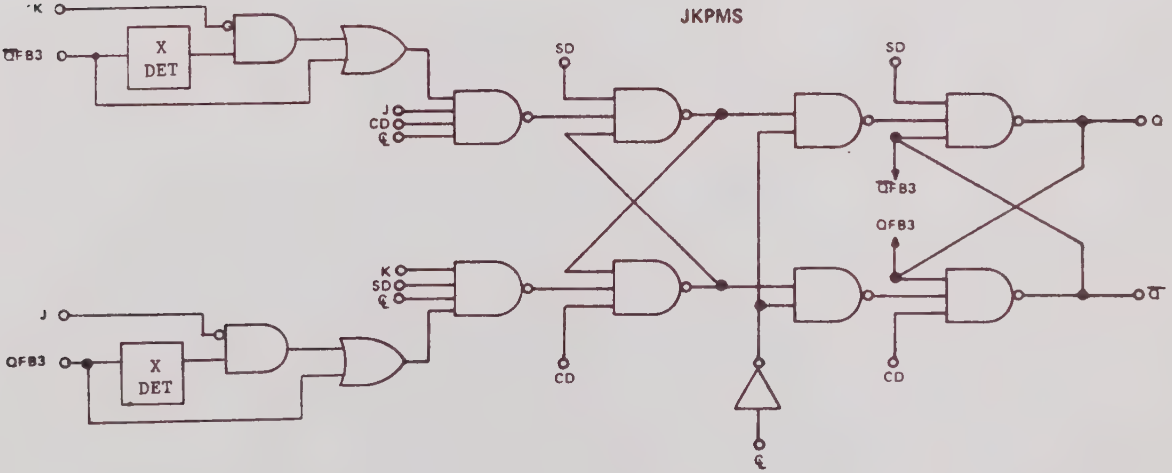

SET
3
SD
J
1
J
Q
6
Q
JKPMS
CLOCK
5
C
K
2
K
Q̄
7
Q̄
CD
4
CLEAR

NOTES

CLOCK	- Master is loaded when clock is HIGH.
J-K Inputs	- If J and K = 0 the FF does NOT change state. If J and K = 1 the FF toggles. If J = 0 and K = 1 the FF resets (Q = 0). If J = 1 and K = 0 the FF sets.
SET	- Low Input to SD sets Q High.
CLEAR	- Low Input to CD sets Q Low.

The JKPMS FF is equivalent to a 7476

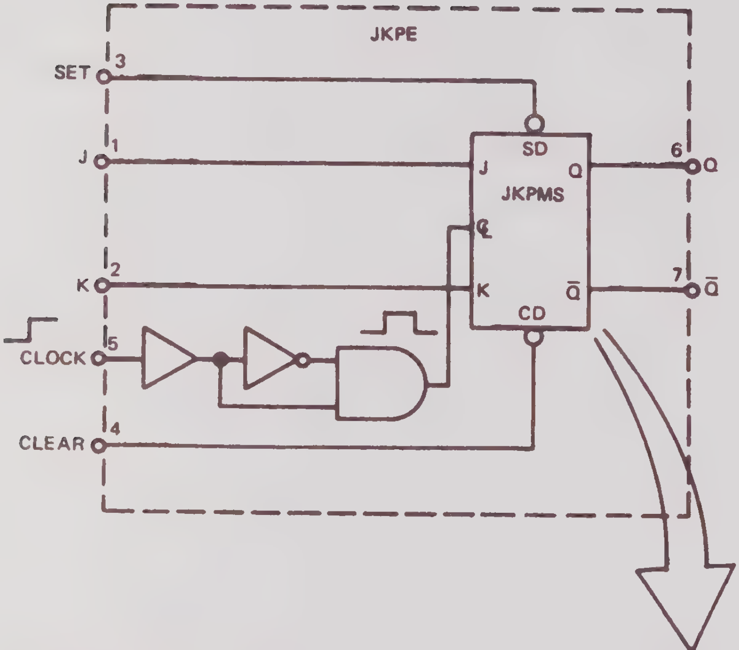
The JKPMS is modeled as:


K
Q̄FB3
X DET
JKPMS
SD
SD
J
CD
C
Q
Q̄FB3
QFB3
K
SD
C
Q̄
J
QFB3
X DET
CD
CD
C

J-K POSITIVE EDGE TRIGGERED FLIP-FLOP

JKPE

The JKPE is modeled as:


JKPE
SET 3
J 1
K 2
CLOCK 5
CLEAR 4
SD
J
Q
JKPMS
C
K
Q̄
CD
6 Q
7 Q̄

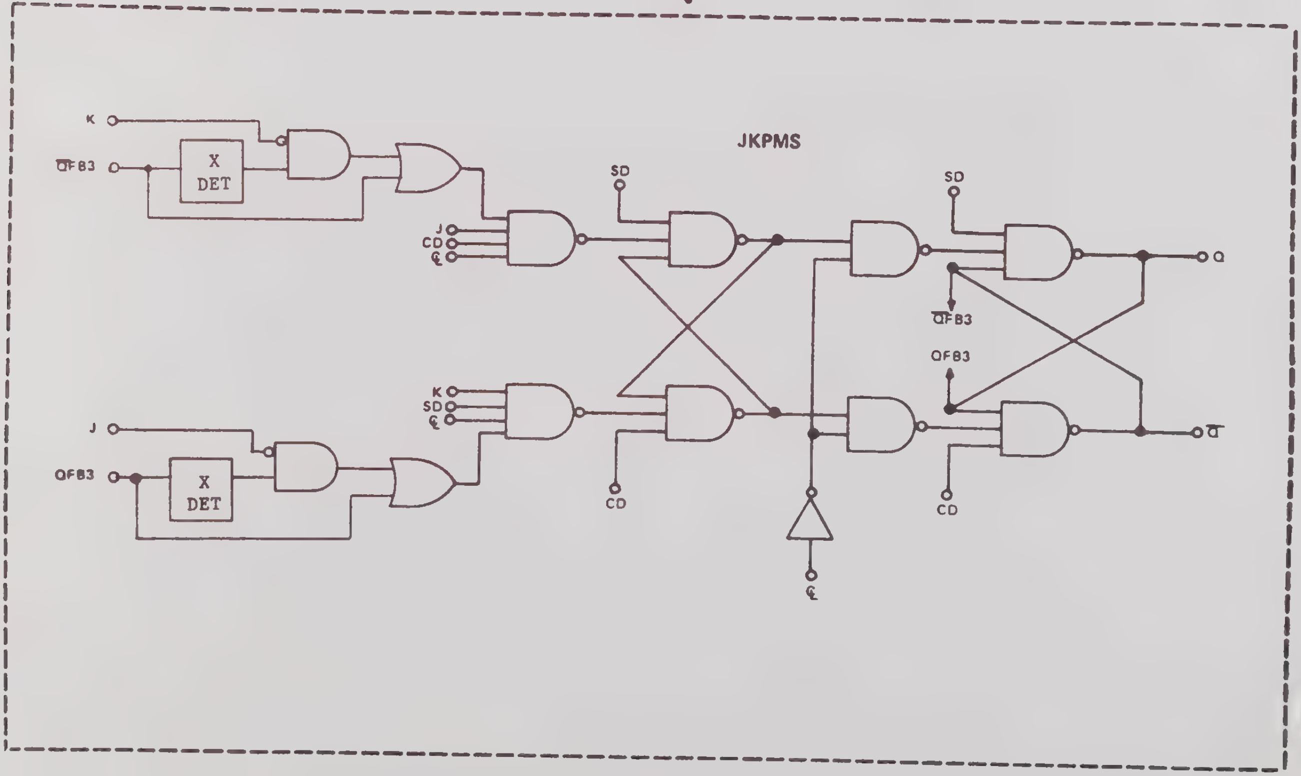
NOTES

CLOCK FLIP-FLOP IS LOADED AS CLOCK GOES HIGH.

J-K Inputs – If J and K = 0 the FF does NOT change state.
If J and K = 1 the FF toggles.
If J = 0 and K = 1 the FF resets (Q = 0).
If J = 1 and K = 0 the FF sets.

SET – Low Input to SD sets Q High.

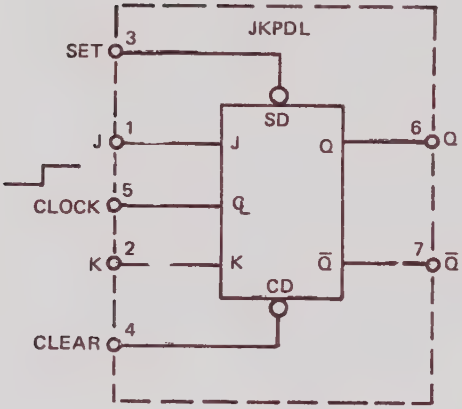
CLEAR – Low Input to CD sets Q Low.


JKPMS
K
Q̄FB3
X DET
J
CD
C
SD
K
SD
C
J
QFB3
X DET
SD
CD
SD
CD
Q̄FB3
QFB3
C
Q
Q̄

J-K FLIP-FLOP WITH POSITIVE DATA LOCKOUT

JKPDL

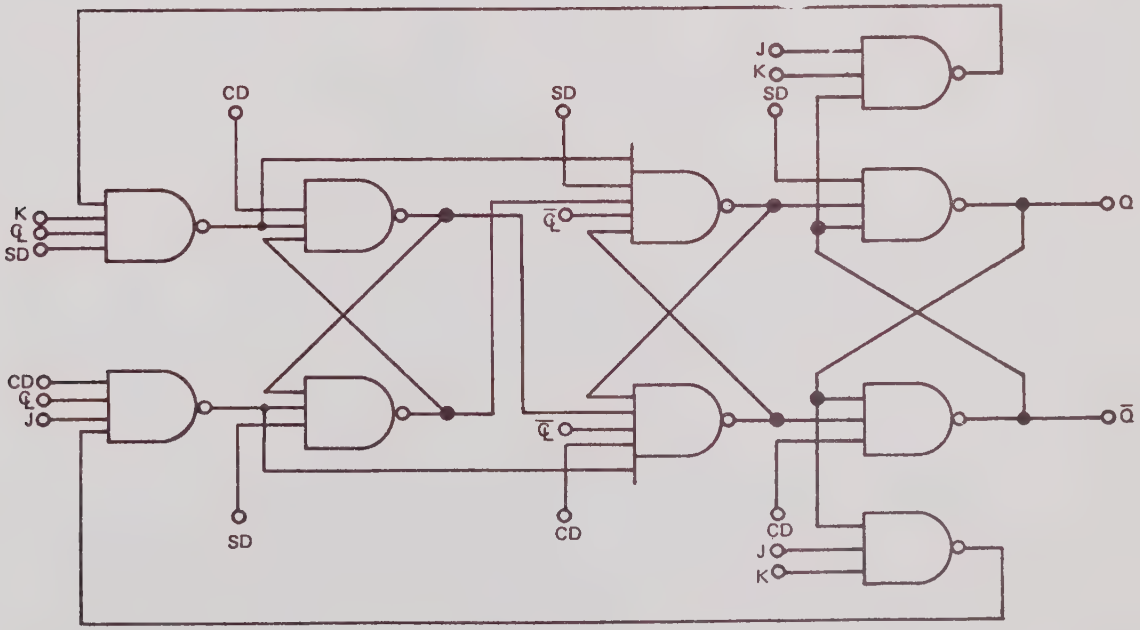
PIN ASSIGNMENTS

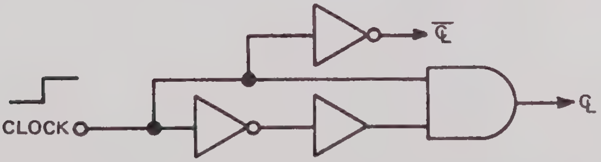

JKPDL
SET 3
J 1
CLOCK 5
K 2
CLEAR 4
SD
J
Q
CL
K
$\overline{Q}$
CD
6 Q
7 $\overline{Q}$

NOTES

CLOCK	– J-K Inputs are enabled only on rising edge of clock pulse.
J-K Inputs	– If J and K = 1 the FF toggles. If J = 1 and K = 0 the FF sets. If J = 0 and K = 1 the FF resets.
SET	– Low Input to SD sets Q High.
CLEAR	– Low Input to CD sets Q Low.

The JKPDL is modeled as:

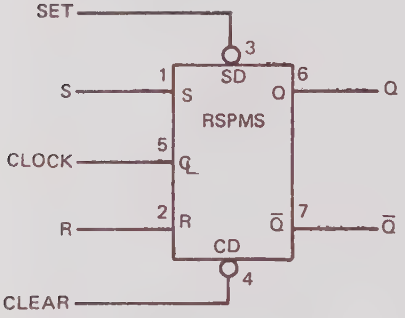

K
CL
SD
CD
CL
J
CD
SD
SD
$\overline{CL}$
$\overline{CL}$
CD
J
K
SD
CD
J
K
Q
$\overline{Q}$


CLOCK
$\overline{CL}$
CL

R-S POSITIVE MASTER-SLAVE FLIP-FLOP

RSPMS

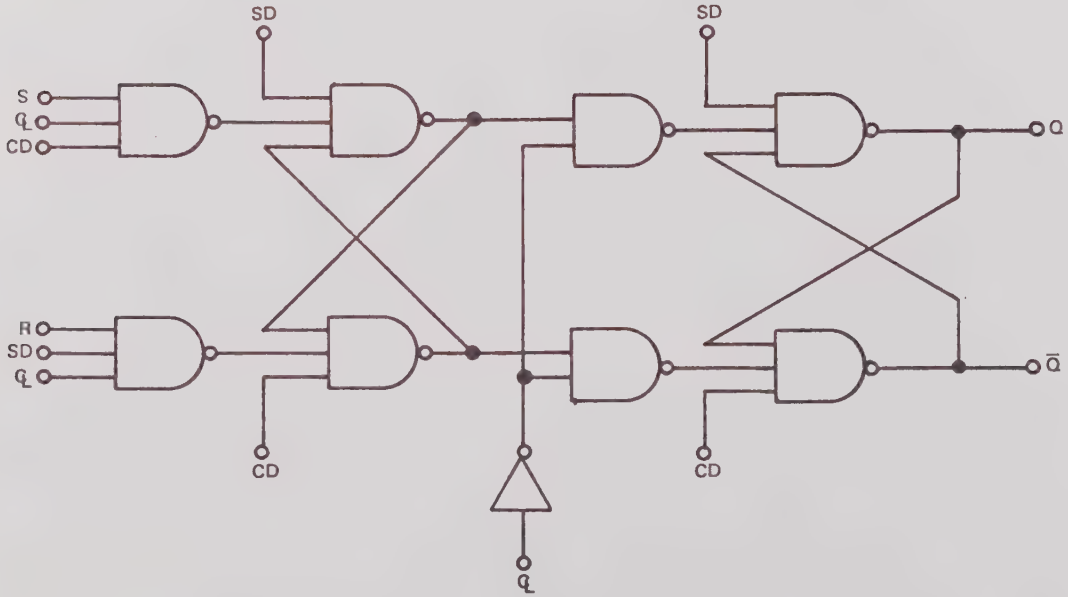
PIN ASSIGNMENTS


SET
3
SD
1
S
6
Q
RSPMS
5
CLOCK
2
R
7
Q̄
CD
4
CLEAR

NOTES

CLOCK	– Master is loaded when clock is HIGH.
R-S Inputs	– If S and R = 0 the FF does NOT change state. If S and R = 1 the FF enters an X state. If S = 0 and R = 1 the FF resets (Q = 0). If S = 1 and R = 0 the FF sets.
SET	– Low Input to SD sets Q High.
CLEAR	– Low Input to CD sets Q Low.

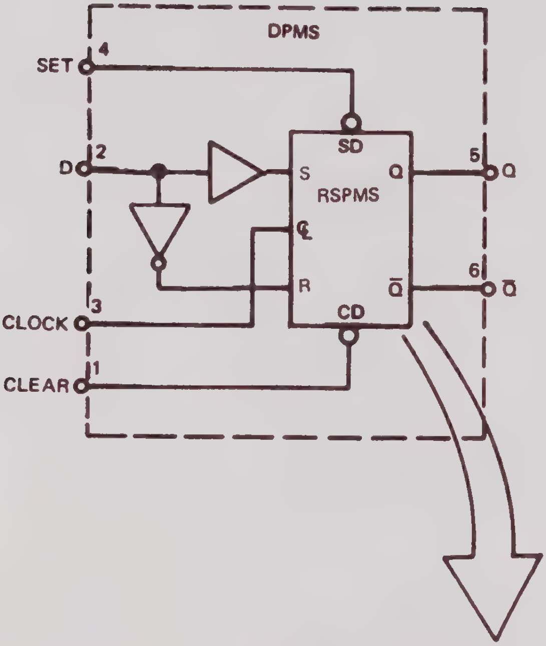
The RSPMS is modeled as:


SD
SD
S
CD
Q
R
SD
Q̄
CD
CD

D TYPE POSITIVE MASTER SLAVE FLIP-FLOP

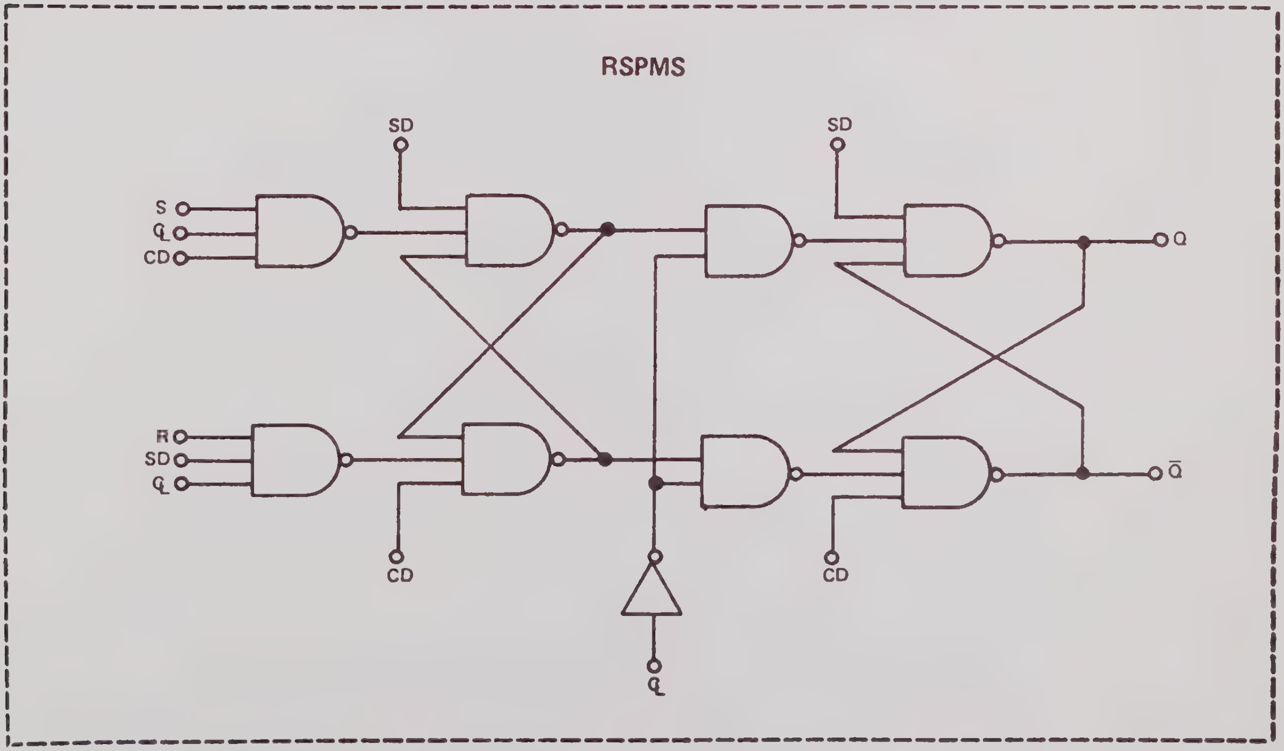
DPMS

The DPMS is modeled as:


DPMS
SET
4
D
2
CLOCK
3
CLEAR
1
SD
S
Q
RSPMS
CL
R
Q̄
CD
5
Q
6
Q̄

NOTES

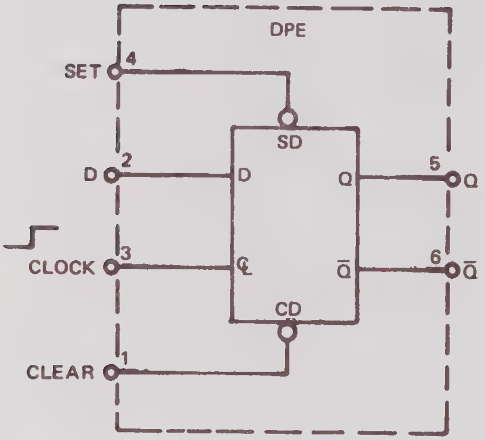
CLOCK	– Master is loaded when clock is HIGH.
D Input	– If D = 0, then S = 0 and R = 1 and the FF resets (Q = 0). If D = 1, then S = 1 and R = 0 and the FF sets.
SET	– Low Input to SD sets Q High.
CLEAR	– Low Input to CD sets Q Low.


RSPMS
S
CL
CD
SD
SD
Q
R
SD
CL
CD
CD
Q̄
CL

D TYPE POSITIVE EDGE-TRIGGERED FLIP-FLOP

DPE

PIN ASSIGNMENTS


DPE
SET
4
SD
D
2
D
Q
5
Q
CLOCK
3
C
$\bar{Q}$
6
$\bar{Q}$
CD
CLEAR
1

NOTES

CLOCK – Input data at D Input is transferred to Q Output on positive edge of clock pulse.

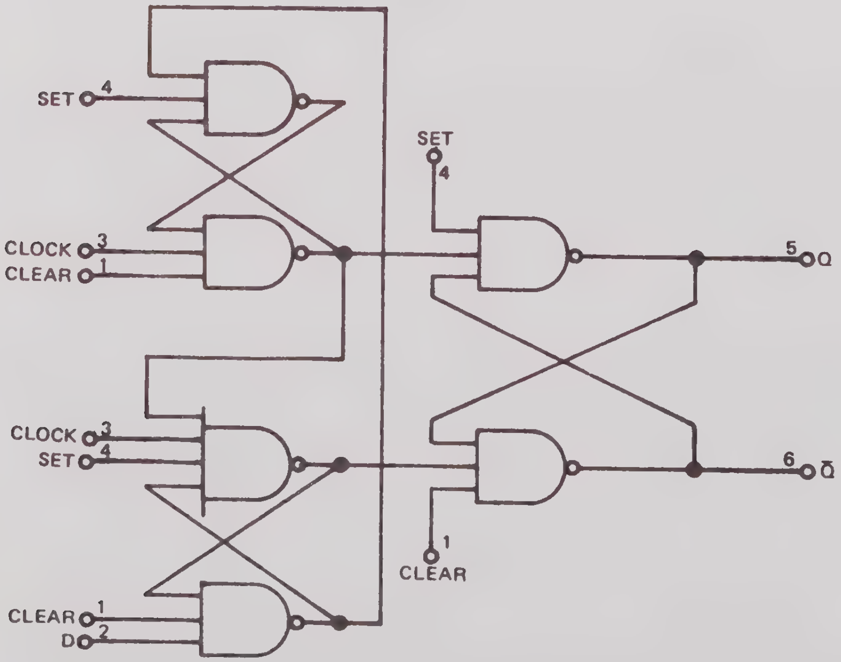
D Input – If D = 0, Q = 0. If D = 1, Q = 1.

SET – Low Input to SD sets Q High.

CLEAR – Low Input to CD sets Q Low.

The DPE is equivalent to a 7474.

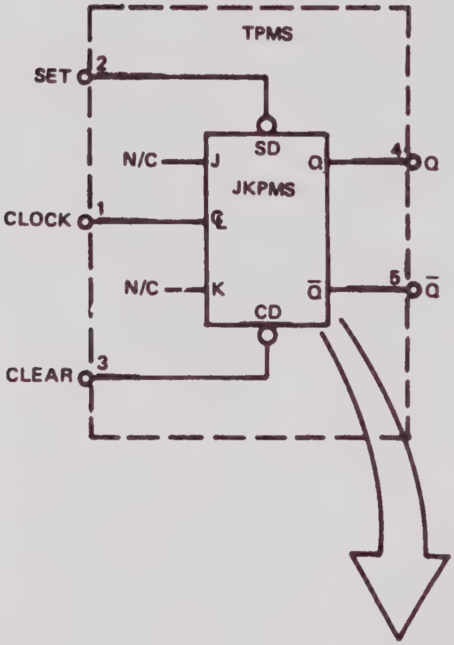
The DPE is modeled as:


SET
4
CLOCK
3
CLEAR
1
CLOCK
3
SET
4
CLEAR
1
D
2
SET
4
5
Q
6
$\bar{Q}$
1
CLEAR

TOGGLE POSITIVE MASTER SLAVE FLIP-FLOP

TPMS

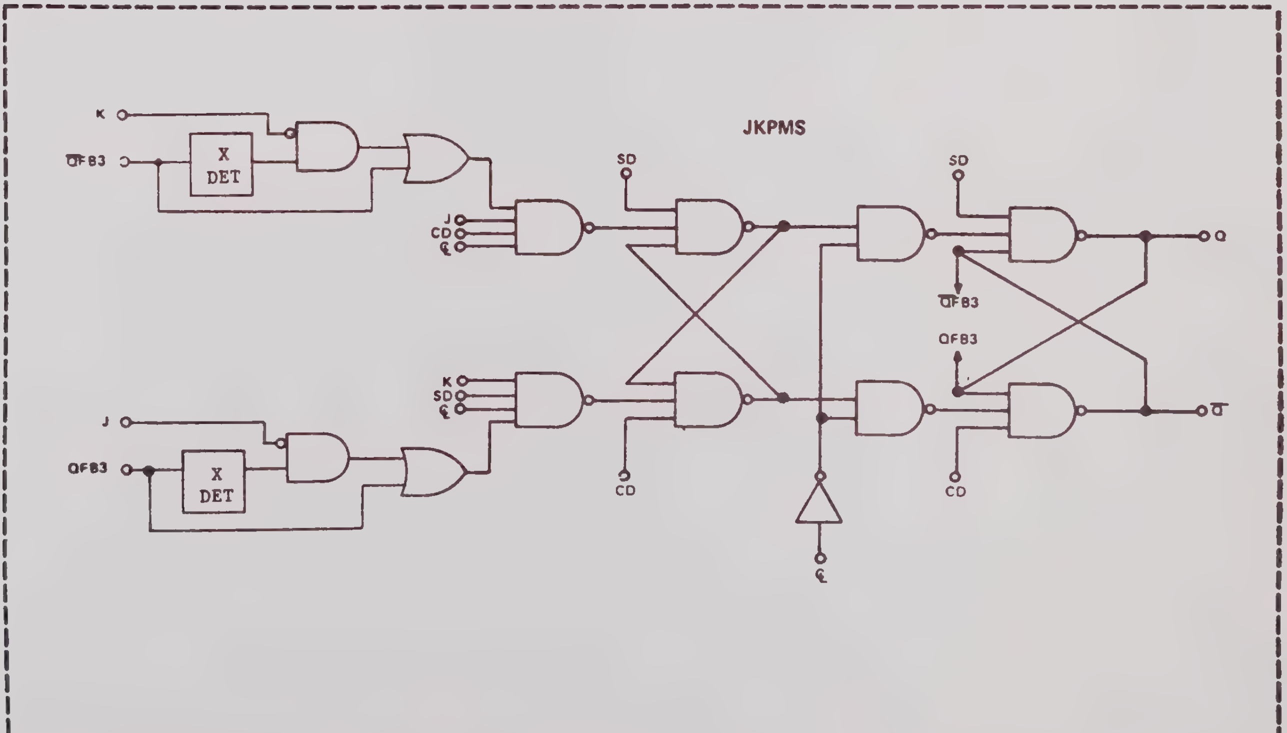
The TPMS is modeled as:



TPMS
SET
2
N/C
J
SD
Q
JKPMS
4
Q
CLOCK
1
C
N/C
K
Q̄
5
Q̄
CD
CLEAR
3

NOTES

CLOCK	– Master is loaded when clock is High. The FF Toggles as the clock goes low (J and K = 1).
SET	– Low Input to SD sets Q High.
CLEAR	– Low Input to CD sets Q Low.

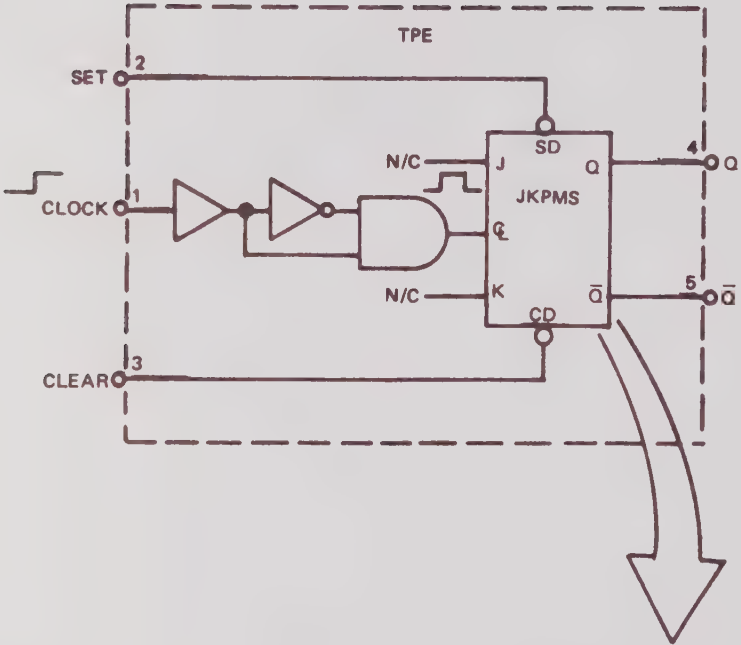


K
Q̄FB3
X DET
JKPMS
SD
SD
J
CD
C
Q
Q̄FB3
QFB3
K
SD
C
J
QFB3
X DET
CD
CD
C
Q̄

TOGGLE POSITIVE EDGE TRIGGERED FLIP-FLOP

TPE

The TPE is modeled as:

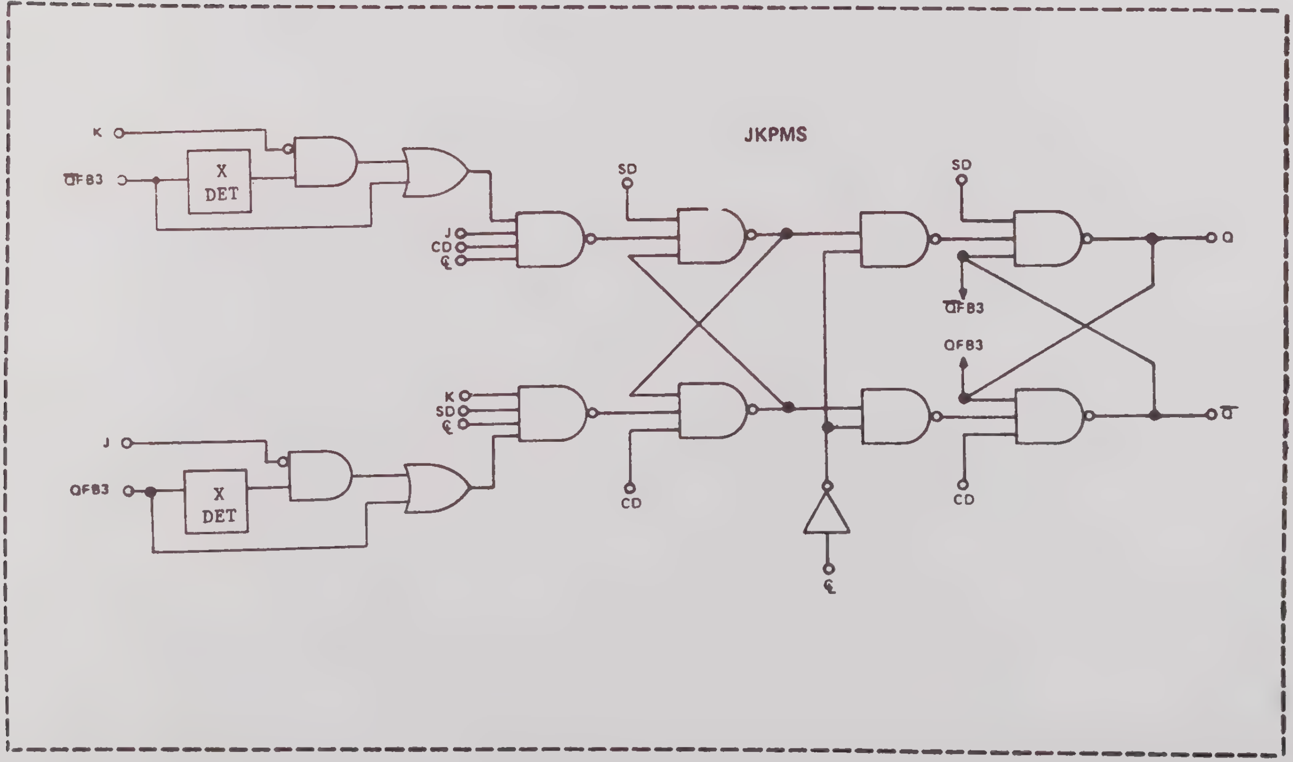

TPE
SET
2
CLOCK
1
CLEAR
3
N/C
J
SD
Q
JKPMS
C
N/C
K
CD
Q̄
4
Q
5
Q̄

NOTES

CLOCK – Master is loaded when clock is High. The FF Toggles as the clock goes low (J and K = 1).

SET – Low Input to SD sets Q High.

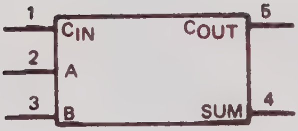
CLEAR – Low Input to CD sets Q Low.


JKPMS
K
QFB3
X DET
J
CD
C
SD
K
SD
C
J
QFB3
X DET
SD
CD
SD
CD
QFB3
QFB3
C
Q
Q̄

n BIT BINARY ADDERS

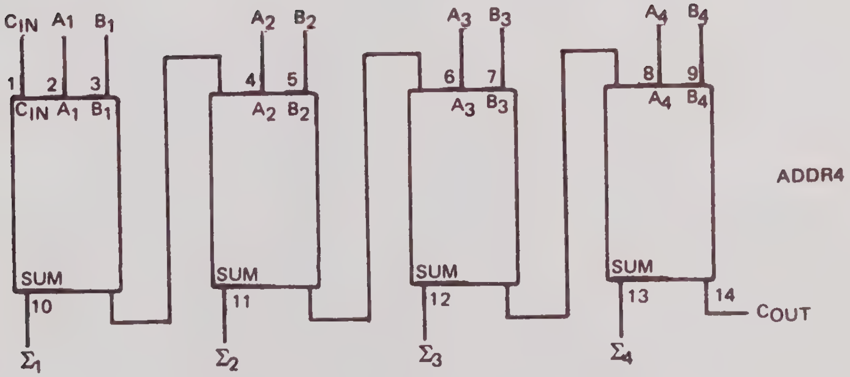
ADDRn (n bit adder)

ADDR1 (1 Bit Adder)


1
CIN
2
A
3
B
COUT
5
SUM
4

$C_{out} = A \bullet B + A \bullet C_{in} + B \bullet C_{in}$
$SUM = A \oplus B \oplus C_{in}$

ADDR4 (4 Bit Adder)


CIN A1 B1
1 2 3
CIN A1 B1
A2 B2
4 5
A2 B2
A3 B3
6 7
A3 B3
A4 B4
8 9
A4 B4
ADDR4
SUM
10
Σ1
SUM
11
Σ2
SUM
12
Σ3
SUM
13
Σ4
14
COUT

ADDER

$\Sigma_1 = A_1 \oplus B_1 \oplus C_{in}$
$\Sigma_2 = A_2 \oplus B_2 \oplus C_1$
$\Sigma_3 = A_3 \oplus B_3 \oplus C_2$
$\Sigma_4 = A_4 \oplus B_4 \oplus C_3$

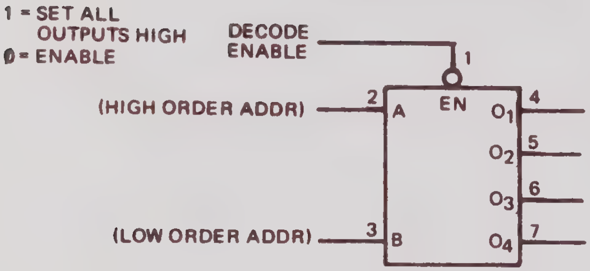
$C_1 = A_1 \bullet B_1 + A_1 \bullet C_{in} + B_1 \bullet C_{in}$
$C_2 = A_2 \bullet B_2 + A_2 \bullet C_1 + B_2 \bullet C_1$
$C_3 = A_3 \bullet B_3 + A_3 \bullet C_2 + B_3 \bullet C_2$
$C_{out} = A_4 \bullet B_4 + A_4 \bullet C_3 + B_4 \bullet C_3$

BINARY DECODER

DECn (n Bit decoder)

DEC2 (2 Bit Decoder)

BINARY DECODER


1 = SET ALL
OUTPUTS HIGH
0 = ENABLE
DECODE ENABLE
1
(HIGH ORDER ADDR)
2
A
EN
B
3
(LOW ORDER ADDR)
O1
4
O2
5
O3
6
O4
7

$O_1 = EN + \overline{\overline{A} \bullet \overline{B}}$

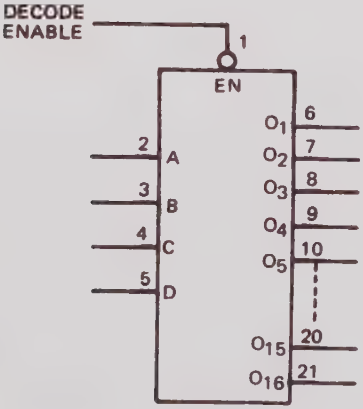
$O_2 = EN + \overline{\overline{A} \bullet B}$

$O_3 = EN + \overline{A \bullet \overline{B}}$

$O_4 = EN + \overline{A \bullet B}$

When enabled, the decoded output is low.

DEC4 (4 Bit Decoder)


DECODE ENABLE
1
EN
2
A
3
B
4
C
5
D
O1
6
O2
7
O3
8
O4
9
O5
10
O15
20
O16
21

$O_1 = EN + \overline{\overline{A} \bullet \overline{B} \bullet \overline{C} \bullet \overline{D}}$

$O_2 = EN + \overline{\overline{A} \bullet \overline{B} \bullet \overline{C} \bullet D}$

•

•

•

$O_{16} = EN + \overline{A \bullet B \bullet C \bullet D}$

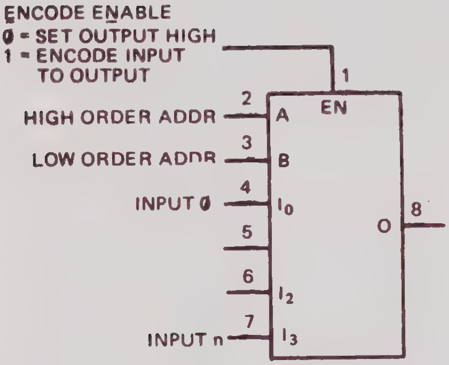
PIN ASSIGNMENTS

PIN	FUNCTION	
1	DECODE ENABLE	
2	DECODE A	DECn
n+1	DECODE n	
(n+1)+1	OUT1 (ADDRESS 0)	
(n+1)+2N	OUTn (ADDRESS 2N 1)	

BINARY ENCODER

ENCn (Encoder with n Bit address lines)

ENC2 (2 Bit Address Lines)

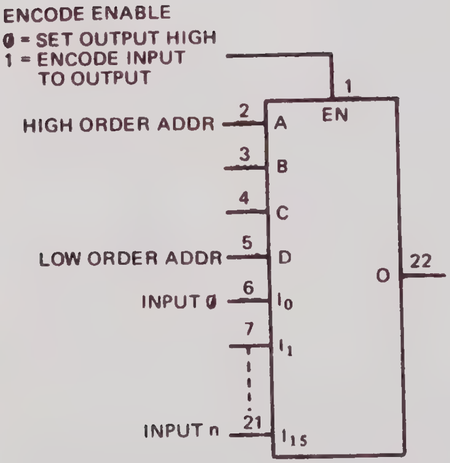

ENCODE ENABLE
0 = SET OUTPUT HIGH
1 = ENCODE INPUT
TO OUTPUT
1
EN
HIGH ORDER ADDR
2
A
LOW ORDER ADDR
3
B
INPUT 0
4
I0
5
6
I2
INPUT n
7
I3
O
8

$$
\begin{aligned}
O = \overline{EN} + (\, & I_0 \bullet \overline{A} \bullet \overline{B} \\
& + I_1 \bullet \overline{A} \bullet B \\
& + I_2 \bullet A \bullet \overline{B} \\
& + I_3 \bullet A \bullet B \,)
\end{aligned}
$$

ENC2							
ADDR			DATA INPUTS				OUTPUT
B	A	EN	I_0	I_1	I_2	I_3	0
0	0	1	S				I_0
0	1	1		S			I_1
1	0	1			S		I_2
1	1	1				S	I_3

S = Input Selected

ENC4 (4 Bit Address Lines)


ENCODE ENABLE
0 = SET OUTPUT HIGH
1 = ENCODE INPUT
TO OUTPUT
1
EN
HIGH ORDER ADDR
2
A
3
B
4
C
LOW ORDER ADDR
5
D
INPUT 0
6
I0
7
I1
INPUT n
21
I15
O
22

ENC4

$$
\begin{aligned}
O = \overline{EN} + (\, & I_0 \bullet \overline{A} \bullet \overline{B} \bullet \overline{C} \bullet \overline{D} \\
& + I_1 \bullet \overline{A} \bullet \overline{B} \bullet \overline{C} \bullet D \\
& \bullet \\
& \bullet \\
& + I_{15} \bullet A \bullet B \bullet C \bullet D \,)
\end{aligned}
$$

PIN ASSIGNMENTS

PIN	FUNCTION
1	ENCODE ENABLE
2	ADDRESS (HIGH ORDER)
n+1	ADDRESS (LOW ORDER)
(n + 1) + 1	INPUT 0 (SELECTED BY ADDRESS = 0)
(n + 1) + 2N	INPUT 2^N–1 (SELECTED BY ALL ADDRESS LINES = 1)
(n + 1) + 2N + 1	OUTPUT

SHIFT REGISTER WITH n OUTPUTS

SFTnS or A

PIN ASSIGNMENTS

OUTPUT PINS

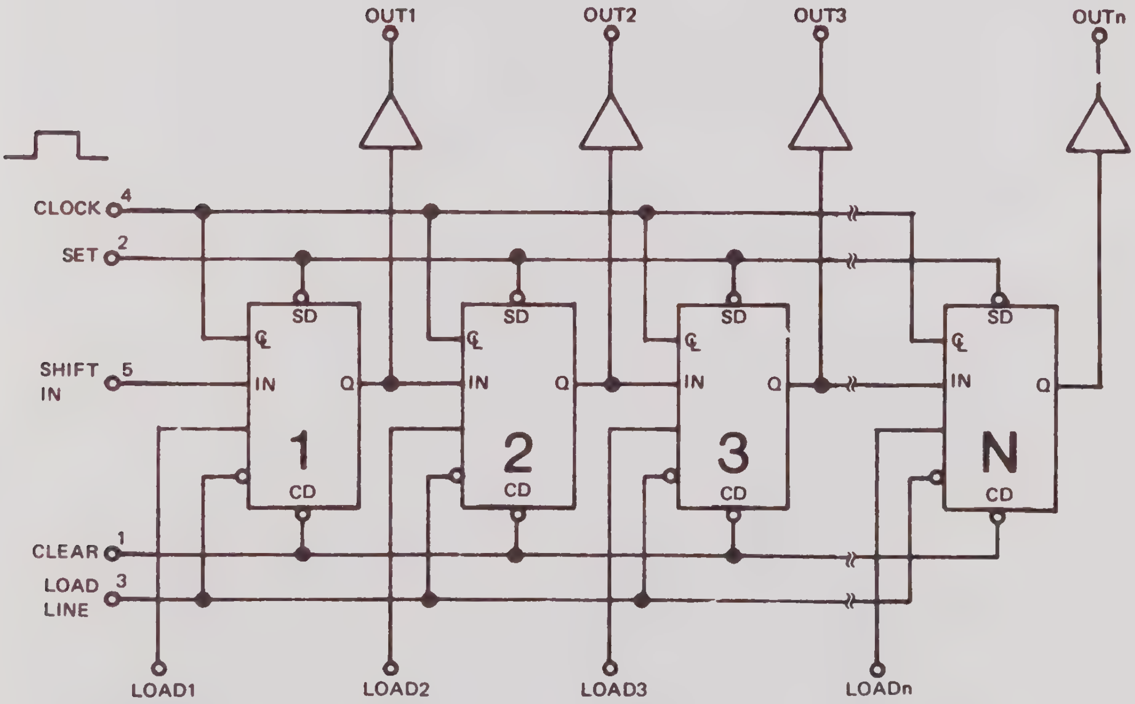
1 + (n + 5) = OUT1
2 + (n + 5) = OUT2
3 + (n + 5) = OUT3
n + (n + 5) = OUTn

LOAD PINS

1 + 5 = LOAD1
2 + 5 = LOAD2
3 + 5 = LOAD3
n + 5 = LOADn

OTHER PINS

PIN	FUNCTION	PULSE
1	CD	‾|_|‾
2	SD	‾|_|‾
3	LOADLINE	‾|_|‾
4	CLOCK	_|‾|_
5	SHIFT IN	


OUT1
OUT2
OUT3
OUTn
CLOCK 4
SET 2
SHIFT IN 5
CLEAR 1
LOAD LINE 3
SD
C
IN
Q
CD
1
2
3
N
LOAD1
LOAD2
LOAD3
LOADn

NOTES

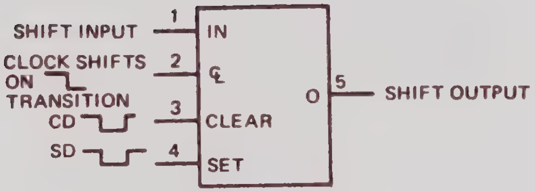
A – Asynchronous operation. Data is CLOCKed IN and OUT only when the LOAD LINE is LOW.

S – Synchronous operation. INput data is shifted IN on leading edge of clock pulse, and shifted OUT on trailing edge.

n · n Output bits.

FAST SHIFT SERIAL IN/SERIAL OUT SHIFT REGISTERS

FSFTn
FSn
FSFn


SHIFT INPUT
1
IN
CLOCK SHIFTS ON TRANSITION
2
C
CD
3
CLEAR
SD
4
SET
O
5
SHIFT OUTPUT

0 AT CLEAR SETS OUT LOW
0 AT SET SETS OUT HIGH
OUT IS UNDEFINED IF CLEAR AND SET ARE BOTH LOW AT THE SAME TIME.

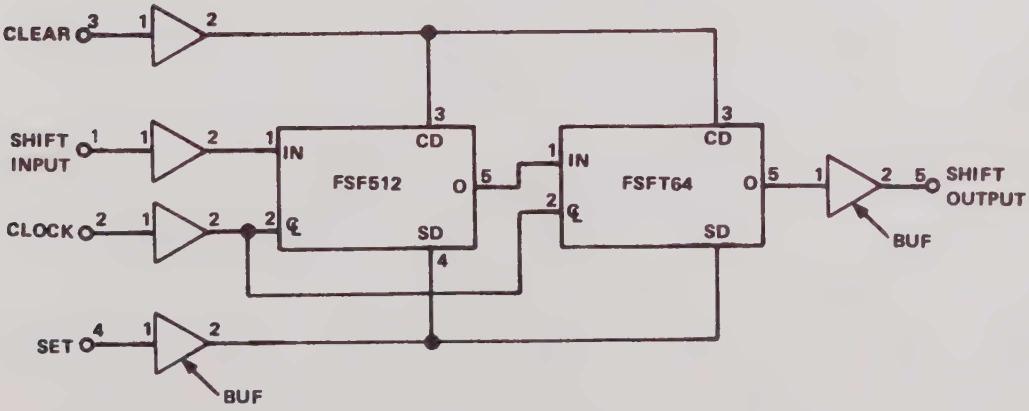
Pin Configuration For FSN, FSFN, and FSFTN

If n=1 – 99 the shift register's name is FSFT, and 4,8,16,25,32,50,60,64, and 80 bit elements are available.

If n= 100-999 the shift register's name is FSF, and 100,128,132,200,250,256,500, and 512 bit elements are available.

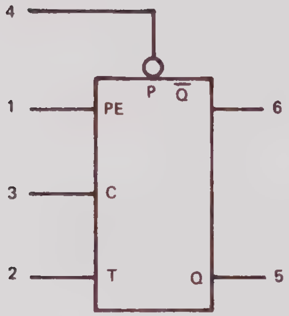
If n= 1000-4000 the shift register's name is FS, and 1000, and 1024 bit elements are available.

Other, longer shift registers can be constructed by cascading elements as in the following example:

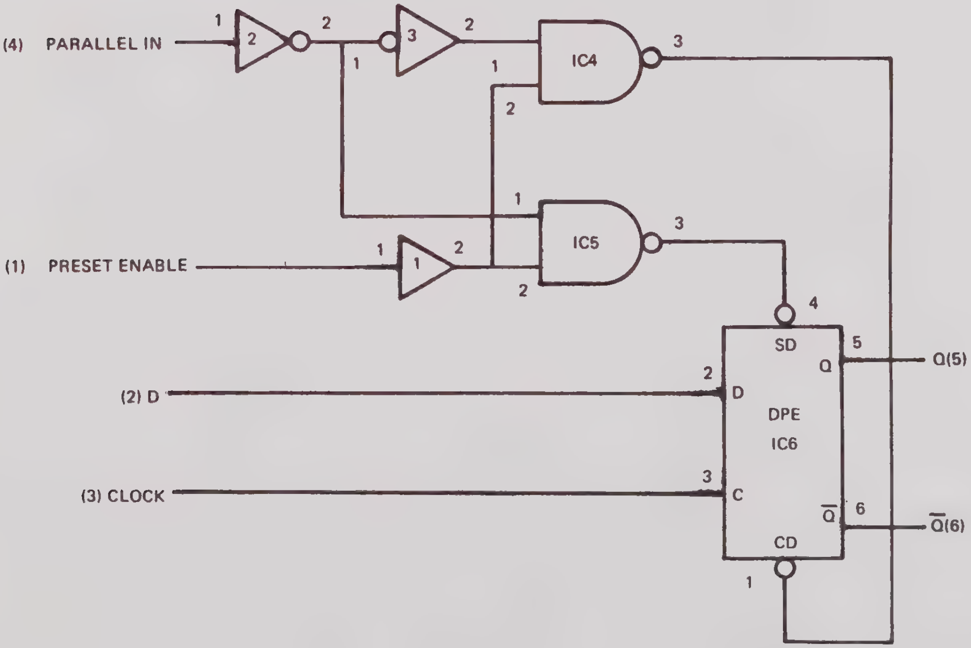

CLEAR
3
1
2
SHIFT INPUT
1
1
2
1
IN
CLOCK
2
1
2
2
C
CD
3
FSF512
O
5
SD
4
SET
4
1
2
BUF
1
IN
2
C
CD
3
FSFT64
O
5
SD
1
2
5
SHIFT OUTPUT
BUF

FSF576

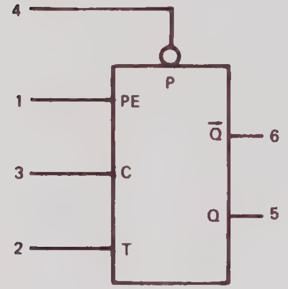
D TYPE POSITIVE EDGE TRIGGERED TOGGLE WITH PARALLEL INPUT
DPEP


4
P
Q̄
1
PE
6
3
C
2
T
Q
5

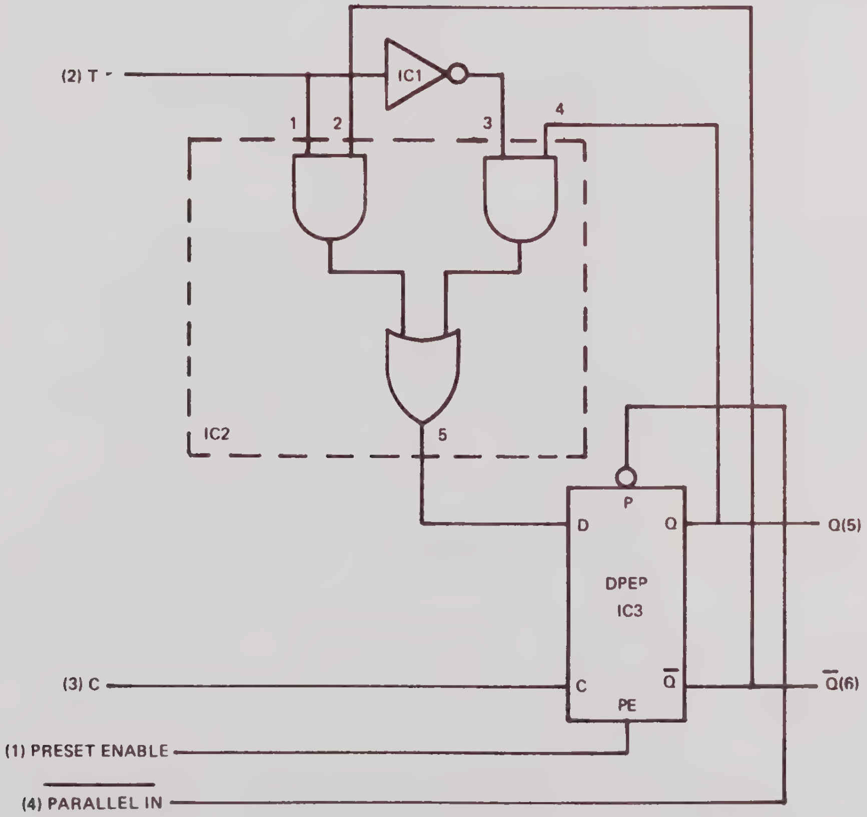
PE	CLOCK	D	Qn+1
1	X	X	PARALLEL IN
0	↑	0	0
0	↑	1	1
0	↓	X	Qn


(4) PARALLEL IN
1
2
2
1
3
2
1
2
IC4
3
1
IC5
3
(1) PRESET ENABLE
1
1
2
2
4
SD
5
Q
Q(5)
2
(2) D
D
DPE
IC6
3
(3) CLOCK
C
Q̄
6
Q̄(6)
CD
1

D TYPE TOGGLE; POSITIVE EDGE, PARALLEL
TPEP

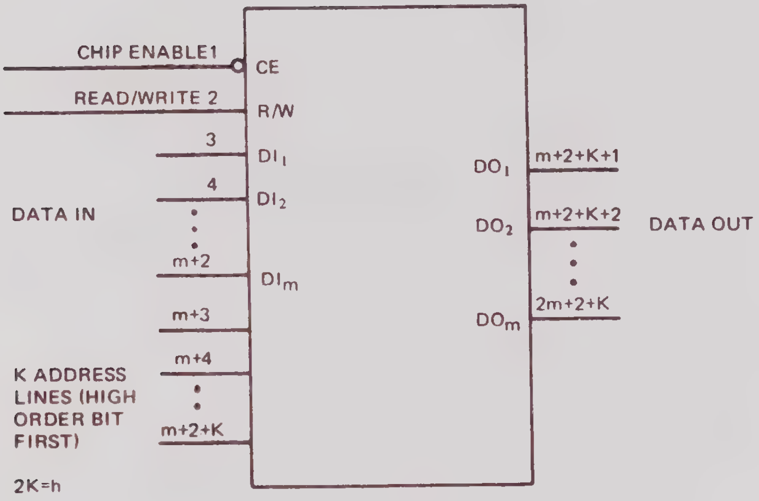

4
P
1
PE
$\overline{Q}$
6
3
C
Q
5
2
T

PE	CLOCK	TOGGLE ENABLE	Q_{n+1}
1	X	X	Parallel In
0	↑	0	Q_n
0	↑	1	$\overline{Q_n}$
0	↓	X	Q_n


(2) T
IC1
1
2
3
4
IC2
5
P
D
Q
DPEP
IC3
C
$\overline{Q}$
PE
Q(5)
$\overline{Q}$(6)
(3) C
(1) PRESET ENABLE
(4) $\overline{\text{PARALLEL IN}}$

RANDOM ACCESS MEMORY

RAMnXm (n by m RAM)


CHIP ENABLE1
CE
READ/WRITE 2
R/W
3
DI_1
4
DI_2
DATA IN
m+2
DI_m
m+3
m+4
K ADDRESS LINES (HIGH ORDER BIT FIRST)
m+2+K
2K=h
DO_1
m+2+K+1
DO_2
m+2+K+2
DATA OUT
2m+2+K
DO_m

A 1 AT CE SETS ALL DO = 1
A 0 AT CE ENABLES RAM

A 0 AT R/W IS READ
A 1 AT R/W IS WRITE

RAMS MAY BE IDENTIFIED AS RM, RAM, OR RR.

RAM'S ARE NOT AS CORE EFFICIENT AS RM'S, BUT RAM'S HAVE FASTER FAULT SIMULATION TIME BY A FACTOR AT MOST OF SIX, BUT TYPICALLY ABOUT THREE.

RR INDICATES RAM'S THAT CAN BE USED AS BUILDING BLOCKS IN RECURSIVE LIBRARY ELEMENTS. (RR DESIGNATED RAMS WILL NOT HAVE THE TIMING ATTRIBUTE).

CURRENT RAM'S AVAILABLE ARE:

RAM 1K x 1 (1024 x 1)
RAM 16 x 1
RAM 16 x 4
RAM 2K x 1 (2048 x 1)
RAM 256 x 1
RAM 256 x 4
RAM 32 x 2
RAM 4 x 2
RAM 4 x 4
RAM 64 x 1
RAM 64 x 9
RAM 8 x 2

BY ONE RAMS (FORCE SIMULATION TO BE 1 IN PARALLEL FOR A FACTOR OF 6 CORE REDUCTION) ARE:

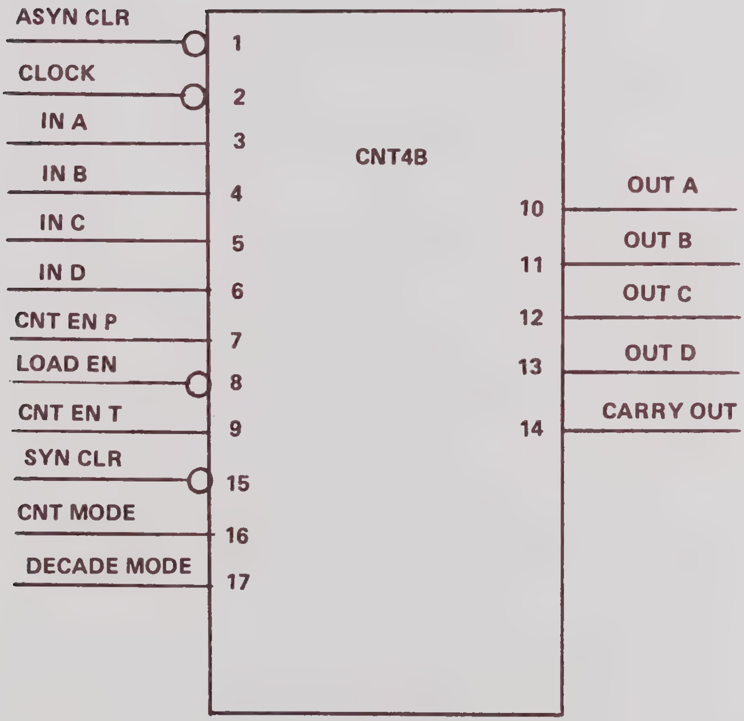
RM1PH FOR H = 256 2K
512 4K
1K

ALL THESE ARE H x 1 RAMS.

DECADE COUNTER
OR
4-BIT BINARY COUNTER

CNT4B

Pin Assignments


ASYN CLR
1
CLOCK
2
IN A
3
IN B
4
IN C
5
IN D
6
CNT EN P
7
LOAD EN
8
CNT EN T
9
SYN CLR
15
CNT MODE
16
DECADE MODE
17
CNT4B
OUT A
10
OUT B
11
OUT C
12
OUT D
13
CARRY OUT
14

CNT4B OPERATING DATA

Input lines operate as follows:

CLOCK	Clearing, loading, or counting affects the masters when the clock goes low, and are transferred to the Outputs as the clock goes high.
ASYN CLR	Low state sets all Outputs (OUTA - OUTD) low regardless of the clock state.
SYN CLR	Low state sets all Outputs (OUTA - OUTD) low after the next clock pulse.
LOAD EN	Low state enables loading of IN A - IN D Inputs.
CNT EN P CNT EN T	Both Count Enable Parallel and Count Enable Trickle must be high to count.
CNT MODE	Selects the type counter desired (decade or binary) as follows: 0 = Decade Counter 1 = Binary Counter
DECADE MODE	Selects the type of Decade device as follows: 0 = Fairchild devices 1 = Texas Instruments or Signetics devices

When established as a Decade Counter (CNT MODE = 0), the following data applies:

If the DECADE MODE line is low, then:

$$\text{CARRY OUT} = \text{CNT EN T} \cdot \text{OUTA} \cdot \overline{\text{OUTB}} \cdot \overline{\text{OUTC}} \cdot \text{OUTD}$$

Under these conditions, the count may be loaded to any state, but will return to its normal sequence within two clock pulses if loaded beyond a count of 9. The homing sequence is:

10,11,4   12,13,4   14,15,0

If the DECADE MODE line is high, then:

$$\text{CARRY OUT} = \text{CNT EN T} \cdot \text{OUTA} \cdot \text{OUTD}$$

Under these conditions, the homing sequence (if the count is loaded beyond 9) becomes:

10,11,6   12,13,4   14,15,2

Note that both types of Decade Counters behave identically if the count is below 10.

FIRST IN FIRST OUT (FIFO) BOOT ELEMENT OPERATING DATA

Functional Description

A low (0) on the Master Reset (MR) causes the FIFO to be emptied.

A low (0) on the Input Enable (IE) causes all input processing to be disabled.

A low (0) on the Output Enable (0) causes all output processing to be disabled, and the Data ouputs (QO - QN) are indeterminate (X).

The Data Bits (DO - DN) are entered into the first location when both Input Ready (IR) and Shift In (SHI) are high (1). This causes IR to go low, but data will stay locked in the first location until both IR and SHI are brought low. Then data will propagate to the first (i.e. last) free location of the FIFO, and data will go high indicating readiness to accept the new input. If the memory is full, IR will stay low.

When data has been transferred to the last cell, Output Ready (OR) goes high indicating the presence of valid data at the Data Output pins. The value of the output pins is always that of the last Memory Cell. The transfer of data is initiated when both the Output Ready (OR) output from the device and the Shift Out (SHO) input to the device are high. This cause Output Ready (OR) to go low. The output data, however, is maintained until both OR and SHO are low. Then the contents of the FIFO will shift down causing OR to go high again. If the memory has been emptied, OR will stay low.

A low to high transition on the Parallel Shift In (PSI), causes the initial data inputs to be loaded. On the next low transition on PSI, the data in the first register is copied into the second and the first control register bit is cleared.

A low to high transition on the Parallel Output (PO), clears the last register bit, causing ORO ot go low.

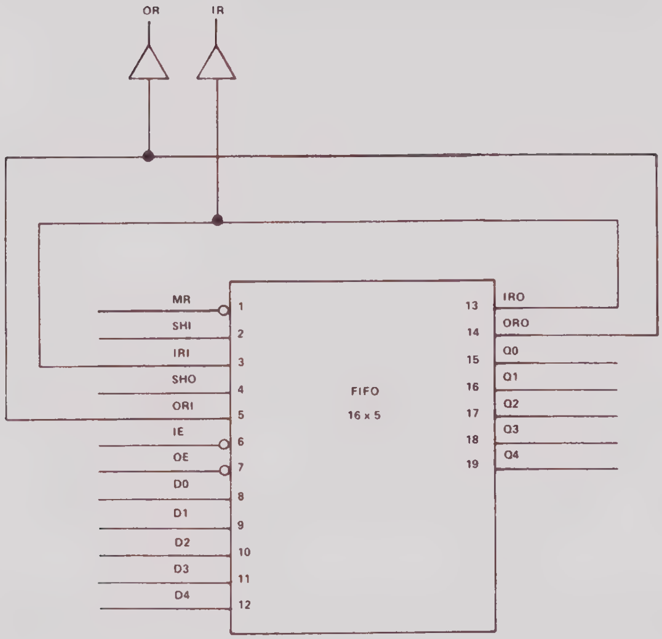
IMPLEMENTATION:

The Input Ready (IRO) pin is to be fed back to the Input Ready Input (IRI) pin, and, similiary for Output Ready.

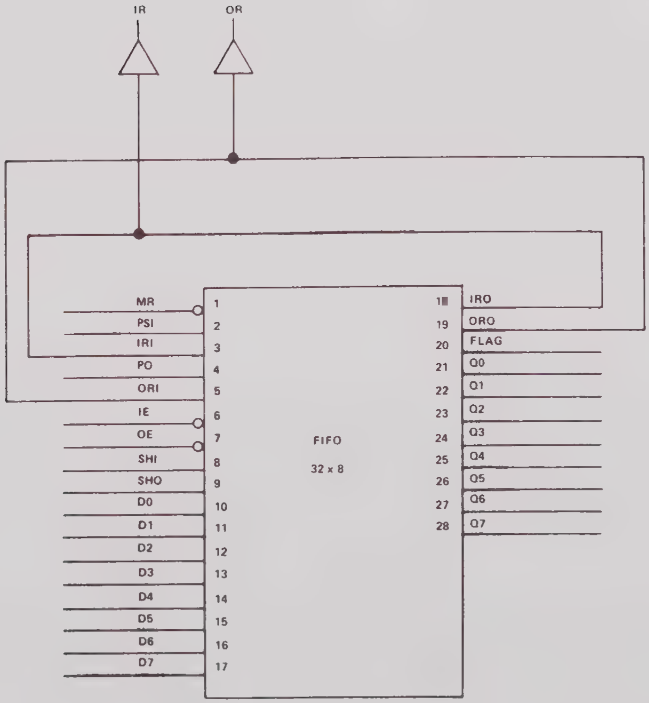
If any Non-Data Input is X (MR, SHI, SHO, IRI, IRO, ORI, IE, or OE) and MR is not low, IRO and ORO are set to X and all data outputs are set to X. This will perpetuate itself until a Master Reset is performed.

MR = Master Reset
DO - DN = Data Inputs
IR = Input Ready
IRO = Input Ready
PO = Parallel Output
IE = Input Enable
QO - QN = Data Outputs
OR = Output Ready
ORO = Output Ready
OE = Output Enable
SHI = Shift In
SHO = Shift Out
PSI = Parallel Shift In

FIFO
16 x 5

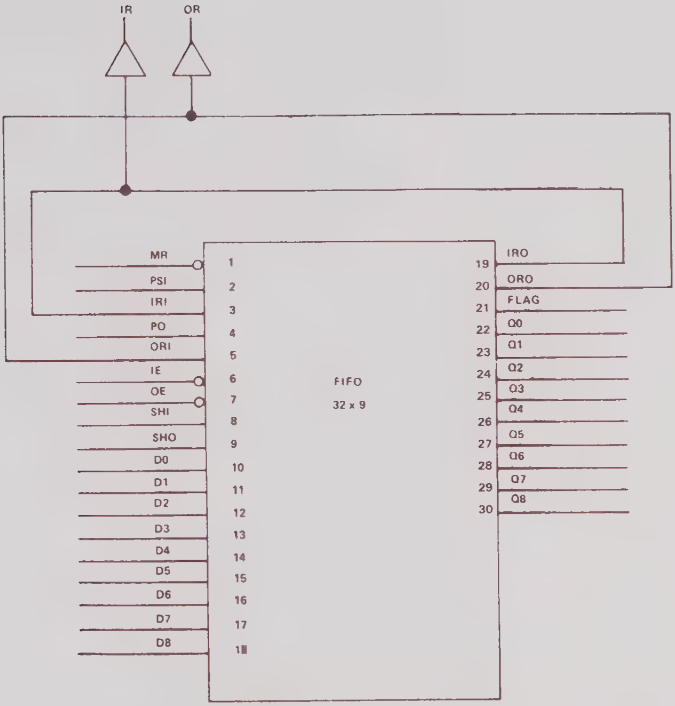

OR
IR
MR
SHI
IRI
SHO
ORI
IE
OE
D0
D1
D2
D3
D4
1
2
3
4
5
6
7
8
9
10
11
12
FIFO
16 x 5
13
14
15
16
17
18
19
IRO
ORO
Q0
Q1
Q2
Q3
Q4

FIFO
32 x 8


IR
OR
MR
PSI
IRI
PO
ORI
IE
OE
SHI
SHO
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
FIFO
32 x 8
18
19
20
21
22
23
24
25
26
27
28
IRO
ORO
FLAG
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7

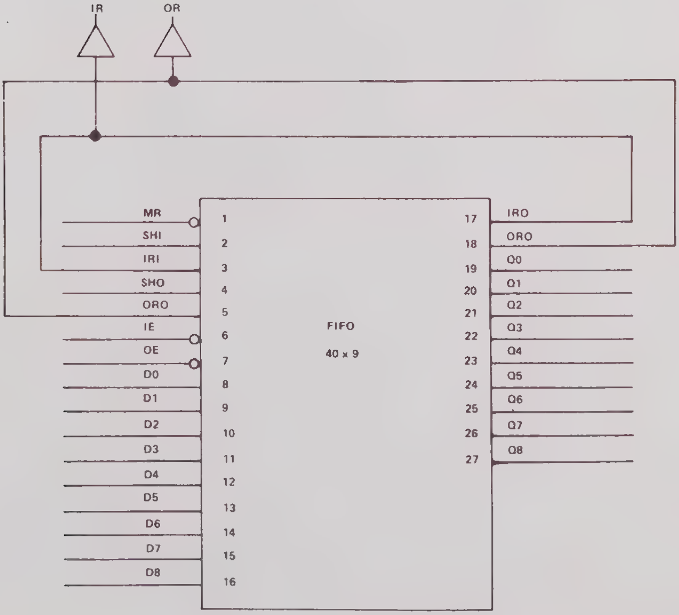
Equivalent to AMD'S 2812

FIFO
32 x 9


IR
OR
MR 1
PSI 2
IRI 3
PO 4
ORI 5
IE 6
OE 7
SHI 8
SHO 9
D0 10
D1 11
D2 12
D3 13
D4 14
D5 15
D6 16
D7 17
D8 1
FIFO
32 x 9
19 IRO
20 ORO
21 FLAG
22 Q0
23 Q1
24 Q2
25 Q3
26 Q4
27 Q5
28 Q6
29 Q7
30 Q8

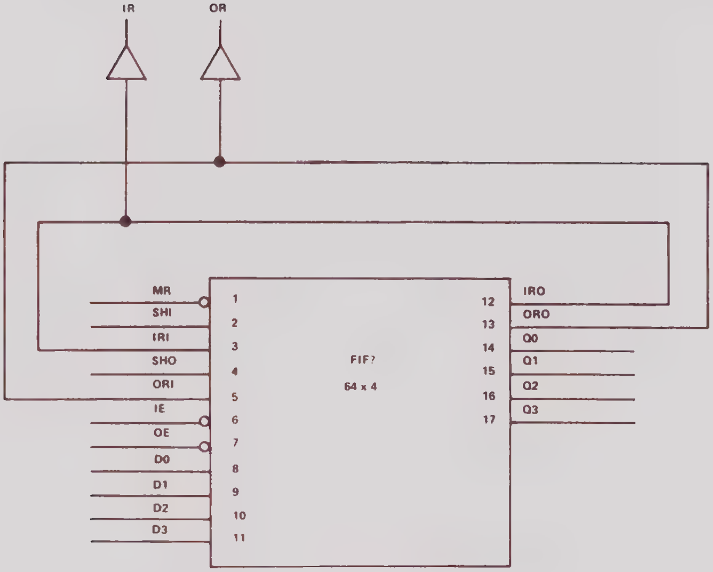
Equivalent to AMD'S 2813

FIFO
40 x 9


IR
OR
MR 1
SHI 2
IRI 3
SHO 4
ORO 5
IE 6
OE 7
D0 8
D1 9
D2 10
D3 11
D4 12
D5 13
D6 14
D7 15
D8 16
FIFO
40 x 9
17 IRO
18 ORO
19 Q0
20 Q1
21 Q2
22 Q3
23 Q4
24 Q5
25 Q6
26 Q7
27 Q8

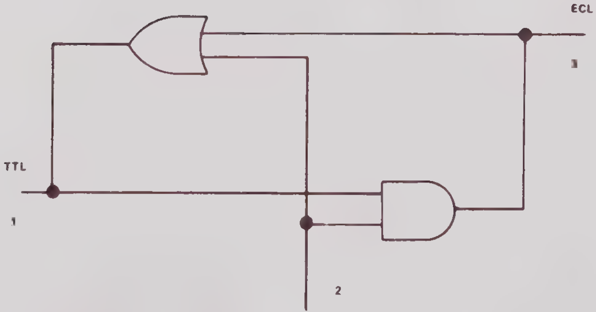
FIF?

64 x 4


IR
OR
MR
SHI
IRI
SHO
ORI
IE
OE
D0
D1
D2
D3
1
2
3
4
5
6
7
8
9
10
11
FIF?
64 x 4
12
13
14
15
16
17
IRO
ORO
Q0
Q1
Q2
Q3

FIFA 64 x 4 AMD
FIFF 64 x 4 - FAIRCHILD
FIFX 64 x 4

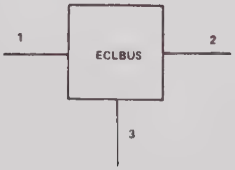
ECLBUS



PIN 1 = D/S
PIN 2 = CONTROL – 0 = SENSE
1 = DRIVE

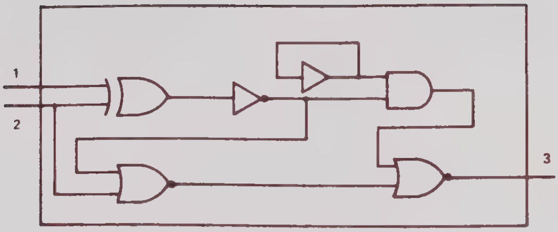
PIN 3 = ECLBUS

PIN TYPES = TTL, TTL, ECL

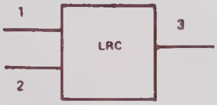


ECL TO TTL CONVERTER
TTL TO ECL CONVERTER

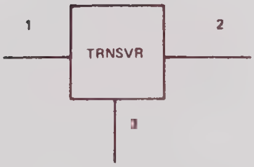
LINE RECEIVER



INPUT 1	0	1	0	1
INPUT 2	0	0	1	1
OUTPUT 3	X	0	1	X



TRANSCEIVER



CONTROL (PIN 3)	0	1
SENSE (PIN 1)	(REC) INPUT	(TRANS) OUTPUT
DRIVE (PIN 2)	(TRANS) OUTPUT	(REC) INPUT

Make Mode–Appendix A

A.1 INTRODUCTION

The CAPS/APG MAKE Mode is used after completing a PREPARE Mode run to transfer the data base files needed for DIAGNOSE Mode operation on 1795/1799 test systems. The CAPS/APG MAKE Mode need only be used under the following set of conditions:

1. If floppy diskettes are being used as system devices in 1795/1799 test systems, then the CAPS/APG MAKE Mode must be run in order to make the necessary floppy diskettes needed for UUT testing in the LOOKUP and/or PROBE sub-modes of the DIAGNOSE Mode.

2. If a hard disk containing the HDS/TEST Monitor only (no CAPS/APG Monitors) is being used on 1795/1799 test systems, it will be necessary to run the CAPS/APG MAKE Mode prior to running the HDS Monitor LOOKUP and/or PROBE sub-modes of the DIAGNOSE Mode for UUT testing.

3. If a hard disk with a CAPS/APG Monitor is being used on 1795/1799 test systems it will not be necessary to run the CAPS/APG MAKE Mode after completing a PREPARE Mode run. The correct DIAGNOSE Mode can be entered for UUT testing directly from a CAPS/APG Monitor without reference to the HDS Monitor.

Whenever the words 1795 are encountered in the following MAKE Mode description, the words 1799 also apply.

A.2 PREPARING CAPS DATA BASE FILES

A.2.1 General

The CAPS X or APG cartridge must be installed as a system device in the top (or only) drive. Normal CAPS runs (for example, a normal PREPARE Mode run) may be accomplished by accessing the CAPS or APG Monitor in the standard manner. For example,

```
.R CAPS ↲        or        .R APG ↲
```

Prior to running the MAKE Mode, it will be necessary to run the D sub-mode of PREPARE with the SETUP Mode set-up for a 1795. Details are contained in Para. A.2.2.

A.2.2 Changes for PREPARE Mode Runs

New boards are run through PREPARE Mode in the same manner as described in the CAPS/APG Manual, except that the user must have previously entered the SETUP Mode and specified ST=FIVE or ST=NINE. Then all of the necessary data base files will be automatically generated as the PREPARE Mode is run. This is the recommended procedure.

If a CAPS X PREPARE Mode run has already been completed without specifying ST=NINE then the user must:

Enter SETUP Mode and specify ST=FIVE (or ST=NINE).

Rerun the D Sub-Mode of PREPARE.

A.3 USING MAKE MODE

A.3.1 General

The MAKE Mode is used to transfer all required files to disks or floppy diskettes. The mode may be entered from the CAPS/APG Monitor using any of the various approaches described in Section 3 of the CAPS/APG Manual. For example,

```
MAKE and XMIT
```

will cause the MAKE Mode to be entered and the MAKE Options page to be displayed (see Figure A-1). Each option on the page may be set as described in the following paragraphs.

A.3.2 FILE USAGE Options

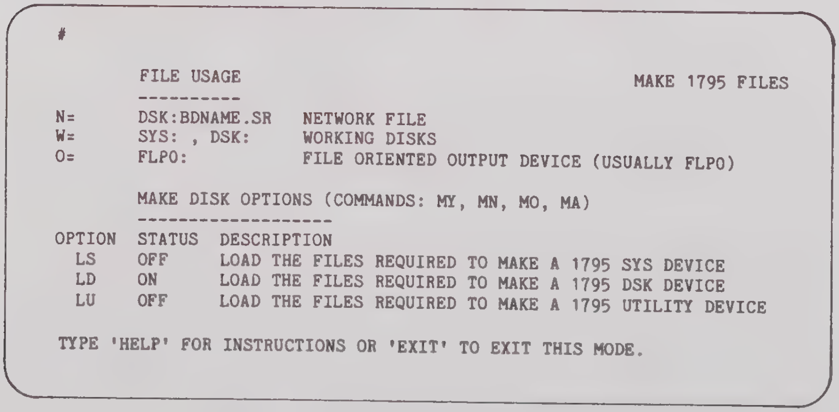
1. NETWORK FILE Name (N=). List the network file name (SR file) and device location of the file. For example,

```
N=DSK2:SAMPLE.SR
```

lists the file for SAMPLE.SR located on DSK2:. If a device is not specified, DSK: is assumed. If a file name extension is not specified, the SR extension is assumed.

2. WORKING DISKS (W=). List the same working devices that were specified during the PREPARE Mode run for the board listed. For example, if DSK0: and DSK1: were specified, then they must also be specified here as the following example indicates.

```
W=DSK0:, DSK1:
```



```
#

        FILE USAGE                                          MAKE 1795 FILES
        ----------
N=      DSK:BDNAME.SR   NETWORK FILE
W=      SYS: , DSK:     WORKING DISKS
O=      FLPO:           FILE ORIENTED OUTPUT DEVICE (USUALLY FLPO)

        MAKE DISK OPTIONS (COMMANDS: MY, MN, MO, MA)
        --------------------
OPTION  STATUS  DESCRIPTION
  LS    OFF     LOAD THE FILES REQUIRED TO MAKE A 1795 SYS DEVICE
  LD    ON      LOAD THE FILES REQUIRED TO MAKE A 1795 DSK DEVICE
  LU    OFF     LOAD THE FILES REQUIRED TO MAKE A 1795 UTILITY DEVICE

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT THIS MODE.
```

Figure A-1. The MAKE 1795 Files Page.

Note that the CAPS files needed by the MAKE Mode options being executed, can only be located and used by MAKE if the W= option is set-up to show their locations. This will occur automatically if W= is set-up for the same working devices used in the PREPARE Mode run that generated the files.

3. FILE ORIENTED OUTPUT DEVICE (O=). This option specifies the disk or floppy output device to be used. If the output is to be to a single floppy drive, this entry will be:

O=FLP0:

However, the following devices may be specified if more than one floppy drive is available;

O=FLP1:
O=FLP2:
O=FLP3:

If the output is to be to a hard disk device, this entry will be:

O=DSK1:	For single disk drives.
O=DSK3:	For dual disk drives.

A.3.3 MAKE DISK Options

The LS, LD, and LU options obtain and load the files required onto a hard disk or a floppy diskette. When the following options are turned ON, the type of device listed (SYS, DSK, or UTILITY) will be made.

Option ON	Makes
LS	Loads the data base and test files required to run the 1795 DIAGNOSIS Mode onto a system device. NOTE This option is not currently used as all needed files are loaded onto the DSK and UTILITY devices through use of the LD and LU options.
LD	Loads the data base and test files required to run the 1795 DIAGNOSIS Mode onto the DSK (non system) device. The files loaded are listed in Table A-1.
LU	Loads the data base files required to run the 1795 PTRANS Mode. Also loads selected source files (that may have to be modified for later runs) onto the device, if available. The files loaded are listed in Table A-1.

The following Sub-Modes of PREPARE must be run prior to transferring files with the MAKE DISK LD Option:

1795 DIAGNOSE Sub-Mode to be enabled:	PREPARE Mode run completed:
LOOKUP	N,T,G,D,F,Q
PROBE/DRM/MANUAL	N,T,G,D

Table A-1

FILES USED TO MAKE 1795 DEVICES

File	Name	Made By	Used By (1795)
	FOR 1795 SYS DEVICE (LS ON)		
None	This option is not currently used (reserved for future expansion).		
	FOR 1795 DSK DEVICE (LD ON)		
QL	1795 Fault Dictionary File	Q PREPARE	LOOKUP
GV	Probe Good Values File	G PREPARE	LOOKUP,PROBE/DRM
SH	1795 Compact .ST File	D PREPARE	LOOKUP,PROBE/DRM
TL	1795 Test Binary File	D PREPARE	ALL MODES
PD	1795 Probe Data Base	D PREPARE	PROBE
LT	Label Table	T PREPARE	PROBE
I3	Interact 3 File	D PREPARE	PROBE
I1	Start Point Data Base	P PREPARE	PROBE (If Available)
I2	Jump Point Data Base	P PREPARE	PROBE (If Available)
I5	User Discr. Data Base	P PREPARE	PROBE (If Available)
I6	Percent Proc. Data Base	P/Z PREPARE	PROBE/DRM
I7	Conditional Connectivity Data Base	P PREPARE	PROBE (If Available)
	FOR 1795 UTILITY DEVICE (LU ON)		
ST	Simulation Table	N PREPARE	PTRANS
LT	Label Table	T PREPARE	PTRANS
DA	CAPS Auto-Prog Test Prog.	G PREPARE	None -Source File
US	User Starting Point File	USER	PTRANS-Source File
UJ	User Jumping Point File	USER	PTRANS-Source File
UD	User Pulse Discr. File	USER	PTRANS-Source File
UC	Conditional Connectivity File	USER	PTRANS-Source File

A.3.4 Running MAKE

The MAKE Mode may be run by installing a diskette or hard disk cartridge in the proper drive, setting-up the desired options on the MAKE page, and then issuing a RUN command using the following set of procedures.

Inserting and Removing Floppy Diskettes

All floppy diskettes must have been formatted prior to initial use in a floppy disk drive. If a new (un-formatted) diskette is being installed, use the FLFORM program to format the diskette after inserting the diskette into the drive (refer to Para. A.4). Proceed as follows to install or remove a diskette in a floppy drive:

Installation:

1. Open the drive door.
2. Insert diskette into drive with index hole and diskette label facing towards the drive door (see Figure A-2).
3. Close drive door to engage spindle and start diskette rotation.

Removal:

1. Open drive door.
2. Remove diskette.

NOTE

If diskette is removed, drive door will not close. Manually depress the door release pin (located inside the door) if another diskette is not to be inserted.

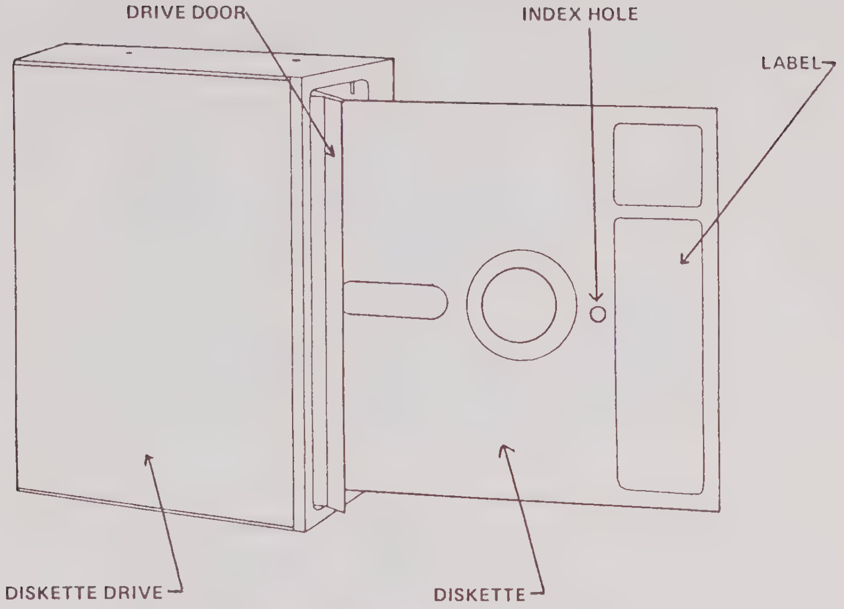


Figure A-2. Diskette Loading.

Making a Diskette or Hard Disk

Proceed as follows to make a diskette or hard disk:

1. Enter the MAKE Mode.

   MAKE and XMIT

2. Set-up the FILE USAGE and MAKE DEVICE options as desired. Figure A-1 shows the page configured to make a 1795 DSK device.

3. Issue a RUN and XMIT command. After completing the run, the monitor top level will again be entered.

A.4 USING FLFORM PROGRAM

The utility program, FLFORM, is used to format a diskette using a soft sectoring technique. The format is 11 sectors of 256 12-bit words per track. There are 77 tracks, therefore, there are a total of 847 sectors containing 216,832 12-bit words per diskette. The program will run on any PDP-8 system containing floppy disk drives. The program is accessed from the OS/8 Keyboard Monitor.

The program first requests which disk drives are to be used for formatting. It then determines if these drives are available. If a drive is not available the system waits until the drive is made available. After all selected drives are made available, formatting is initiated one diskette at a time. After each diskette is formatted, its formatting is verified by checking all sector identifiers. If a formatting error is found, an error message containing the offending track and sector address is transmitted to an output device. After checking all sector addresses the program sends a format complete message and the number of errors found to the output device. If errors are detected the diskette should be reformatted. It then continues on to the next diskette. The program repeats until all specified diskettes are formatted.

NOTES

1. ↑C is recognized at all times and will initiate a return to OS/8. Typing "↑C" while the diskette is being formatted aborts the operation and the diskette will not be completely formatted. The ↑C is echoed and the program halts after printing, "READY SYSTEM DEVICE - PRESS CONT TO RETURN TO OS/8." (On PDP-8A processors the RUN Pushbutton is used.)

2. Write protect and not ready states are checked during each operation. If either is detected an error message ("FLOPPY DISK UNIT X NOT READY") is printed and the operation is retried until the error is corrected.

3. When a DMA channel failure is detected an error message (DMA CHANNEL FAILURE -- PRESS CONT TO RETURN TO OS/8) is printed and the processor halts.

4. If non-recoverable controller errors are detected the drive is disconnected and an error message ("UNIT X HAS ERRORS DISCONNECTED") is printed. If all drives are disconnected another error message (ALL UNITS DISCONNECTED -- PRESS CONT TO RETURN TO OS/8) is printed followed by a processor halt. Readying the System device and depressing the CONT switch returns the operation to OS/8. (On PDP-8A processors the RUN pushbutton is used.)

To run FLFORM perform the following:

1. Enter Keyboard Monitor.

NOTE

When the Keyboard Monitor is ready to accept commands a dot (.) printout occurs.

2. Type .R FLFORM. ⤶
   The following printout should occur:

```
FLOPPY DISK FORMATTER VER.X
SELECT THE UNITS FOR FORMATTING (Y or N)
```

3. Install the floppy diskette to be formatted in the floppy drive selected (normally UNIT 0).

4. Now the operator is asked which diskettes are to be formatted. The operator must answer each question as it is asked.

CAUTION

If UNIT 0 is formatted in a 1795 test system, and the system diskette is installed, it will be formatted and thus destroyed.

PRINTOUT	RESPONSE	COMMENT
UNIT 0 ?	Y or N	Y (yes) to format this diskette.
UNIT 1 ?	Y or N	Same as above.
UNIT 2 ?	Y or N	Same as above.
UNIT 3 ?	Y or N	Same as above.
ARE YOU SURE ?	Y or N	Y (yes) must be specified to format selected diskettes. If N is typed, the program will repeat, asking which drives to use.

NOTE

If on subsequent retries and if the specified drive was previously disconnected an error message is printed-DRIVE PREVIOUSLY DISCONNECTED BECAUSE OF ERRORS-INPUT IGNORED.

FLFORM will format diskettes installed in any drive, (UNITS 0 through 3) by answering Y (yes) to the proper question as it is asked.

After each diskette is formatted the program prints:

```
FORMATTING COMPLETE ON UNIT X
THERE WERE N ERROR DETECTED
```

This message is repeated when each selected diskette is formatted. Once all selected diskettes are formatted the program will restart from the beginning with the message:

```
SELECT DISKS TO BE FORMATTED (Y or N)
```

APG Use–Appendix B

B.1 INTRODUCTION

This appendix includes the procedures necessary to use the Incremental Simulation (I) Sub-Mode of PREPARE and the Equation Generation, Tracking, and Connectivity (E/T/C) Mode. Both the I Sub-Mode of PREPARE and the E/T/C Mode are only supplied with APG-equipped test systems.

B.2 INCREMENTAL SIMULATION

B.2.1 General

Incremental Simulation provides the abilty to monitor the effects of test sequences as they are appended to the test program. Incremental Simulation allows test programs to be generated, sequence by sequence, and the results evaluated with Incremental Automatic Fault Detection (AFD).

With Incremental Simulation, a portion of the test program can be developed and portions appended to it later with few redundant efforts. The immediate availability of simulation results allows the user to focus his attention on testing a board segment until satisfactory results are achieved.

Figure B-1 illustrates the principal steps required to perform Incremental Simulation.

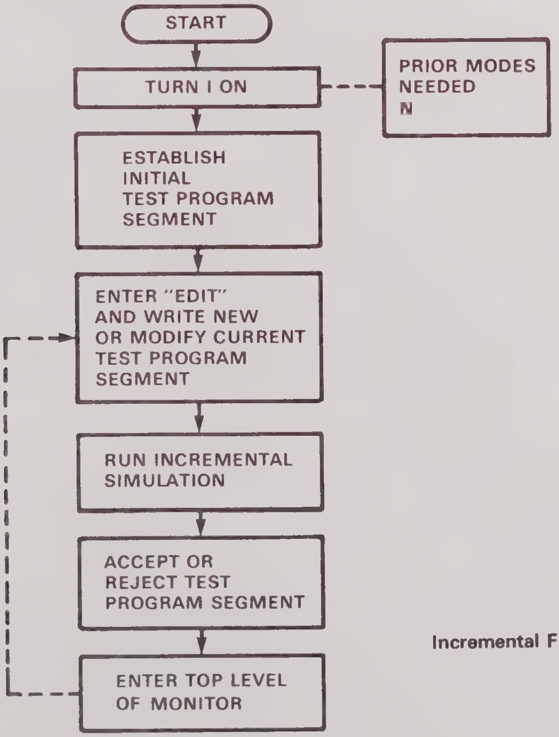

START
TURN I ON
PRIOR MODES NEEDED N
ESTABLISH INITIAL TEST PROGRAM SEGMENT
ENTER "EDIT" AND WRITE NEW OR MODIFY CURRENT TEST PROGRAM SEGMENT
RUN INCREMENTAL SIMULATION
ACCEPT OR REJECT TEST PROGRAM SEGMENT
ENTER TOP LEVEL OF MONITOR

Figure B-1.

Incremental Flow Simulation Flow Chart.

Basically, there are two parts to a test program developed under the Incremental Simulation Mode of PREPARE. They are called the "trial" and "accepted" parts.

The "trial" part of the test program is the segment that is undergoing development (i.e. being written, edited, and processed). The Fault Detection obtained by the test program including the "trial" section can be examined using Incremental AFD. The DISPLAY and ETC Modes can be used to display the simulated board operation and track problems respectively. The "trial" segment can be modified and resimulated until its performance is acceptable. It is then "accepted" and appended to the existing test program. This process is repeated until the total test program is generated.

If at any time it is desired to add more segments to the test program it is only necessary to write, edit, and process that additional information. It is not required to reprocess the previously "accepted" segments of the test program.

B.2.2 Accessing the INCREMENTAL SIMULATION Sub-Mode of PREPARE

To access the I Sub-Mode of PREPARE, enter the PREPARE Mode from the Monitor and turn the EXECUTE Sub-Mode I option ON.

NOTE

> I Sub-Mode supercedes all other Sub-Modes of PREPARE and must be turned off to access any other Sub-Mode of PREPARE.

B.2.3 Incremental Simulation Option Pages

To generate a test program using the Incremental Simulation Sub-Mode of PREPARE the user must specify the options of PREPARE as indicated in the following paragraphs.

NOTE

> Only the PREPARE Mode options that affect Incremental Simulation are listed. Other PREPARE Mode options are not applicable when using the I Sub-Mode.

B.2.3.1 FILE USAGE page

The FILE USAGE page is shown in Figure 5-11. Options on this page that must be set-up to run the I Sub-Mode are as follows:

N=(NETWORK FILE) This is the name assigned to the network description (.SR) file that is to be run in the PREPARE Mode and the device on which the file is located. It is also the name assigned to all files used in the I Sub-Mode except for the Test Program file. Files may be located on any file structured disk-based device. If a device is not specified, the default device

(DSK:) is assumed. If an extension is not specified, the .SR extension is assumed. A file labeled NAMEA.SR and located on DSK: could therefore be specified by the simple command:

```
N=NAMEA
```

If the file had been located instead on DSK2:, the following command would have been necessary:

```
N=DSK2:NAMEA
```

T=(TEST PROGRAM FILE) This is the name assigned to the file which contains the initial segment that is to be run in the I Sub-Mode of PREPARE, and the device on which that file is located. Files may be located on any file structured disk device. If a device is not specified, DSK: is assumed. Any unused extension may be specified, if one is not specified, the .TP extension is assumed. When a "trial" set is "accepted", the T= file is updated with the new .TP.

U=(USER IC LIBRARY FILE) Applicable only to the corresponding N Sub-Mode runs.

LM=(USER LANGUAGE MACRO FILE) The LM= command specifies the archive file name and location where user-generated macros that are called and used within the test program are stored. The archive file specified can reside on any of the disk surfaces and does not have to be on the same surface specified for the test program.

LI=(LISTING DEVICE) Output all Incremental Simulation Sub-Mode listings to the device specified as in the following examples:

LI=LPT:	Output listings to the line printer
LI=TEC:	Output listings to the TEC
LI=TTY:	Output listings to the Teletype

If LI=DSK: is specified, the listing files will be left on the working disk for later reference with LISTALL or LOOK.

W=(WORKING DISKS), X=(EXTENDED WORKING DISKS) CAPS and APG create a series of special-purpose test (W=) and listing files (X=) during a run and automatically store these files on the devices named. Table 5-9 lists the CAPS and APG files created during a PREPARE Mode run, briefly defines their purposes, and lists normal file locations. Note that the usage of DSK0: and DSK1: for incremental simulation is not recommended as there is insufficient space on the system disk to accomodate the necessary intermediate files.

Temporary files listed in the tables are CAPS and APG files directed to the extended working disks (X=). These may be deleted at the end of a sub-mode run or are deleted automatically during the run.

FM=(OUTPUT FORMAT) Selects the proper output format for listings (NARROW, WIDE, or SCOPE) and applies only to the listing files created in the N Sub-Mode or later Sub-Modes of PREPARE.

B.2.3.2 SELECT MISCELLANEOUS OPTIONS page

The SELECT MISCELLANEOUS option page is shown in Figure 5-16. The following option may be set prior to running the I Sub-Mode:

P (PRINT (ALL) ERRORS ON TELETYPE) Normally, error messages are output to the Teletype or to the listing device specified by the FILE USAGE Page LI= Option (depending on the software module in use when the error occurs). If the P Option is ON, all error messages will be output to the Teletype, and will not be printed on the listing device specified by the LI= Option.

L (LONG ERROR MESSAGES) Long error messages define errors in clear text (L ON). When L is OFF, abbreviated error messages are generated. Some examples of long and short error messages are provided in the Pre-Processor Sub-Section.

E (INVOKE EDITOR ON ERRORS) This option allows EDIT to be entered if errors are detected in the source file being processed by APG during a PREPARE Mode run. If E is not active and errors occur, error messages will be generated, execution will be suppressed, and the APG Monitor will be entered. When E is active, EDIT is entered automatically so that the source file being processed can be corrected "on-line". Once corrections have been made, pressing the X and XMIT keys will result in an immediate return to the aborted APG run, and the run will begin again at or near the point of termination. If EDIT is exited via the E and XMIT command, a return to the APG Monitor will take place.

M (RUN MACRO TEST PROGRAM PROCESSOR) This option is not applicable to the I Sub-Mode of PREPARE. (The MACRO TEST PROGRAM PROCESSOR is always run in the I Sub-Mode.)

N (RE-NUMBER TEST PROGRAM MACRO STATEMENTS) This option is not applicable. Renumbering can not be performed in the Incremental Simulation Sub-Mode.

C (WRITE FAULTS BY CLASS NUMBER) This option produces a list of all modeled faults by class number, displayed following Fault Detection runs.

V (WRITE UNDETECTED FAULTS BY CLASS NUMBER) This option produces a list of all modeled faults, indicating which faults remain undetected by the test program. The list is displayed following a Fault Detection run.

I (WRITE DETECTED FAULTS BY EXT AND IC PIN) With this option ON, the listing displayed following a Fault Detection run will include EXTERNAL FAULT DETECTION by EXT and IC pin.

U (WRITE UNDETECTED FAULTS BY EXT AND IC PIN) With this option ON, the listing following a Fault Detection run will also include UNDETECTED EXTERNAL FAULTS by EXT and IC pin.

The remaining options on the Select Miscellaneous option page do not apply to the Incremental Simulation Sub-Mode of PREPARE.

B.2.3.3 PRINT LISTINGS page

The options on this page allow the listings produced by a Fault Detection run to be displayed following the run.

F (FAULT CLASSES MODELLED FOR DETECTION SIMULATION). This option displays the .FP file following a Fault Detection run.

D (DETECTED FAULT CLASSES AND PERCENTAGE DETECTION). This option displays the .FG file following a Fault Detection run.

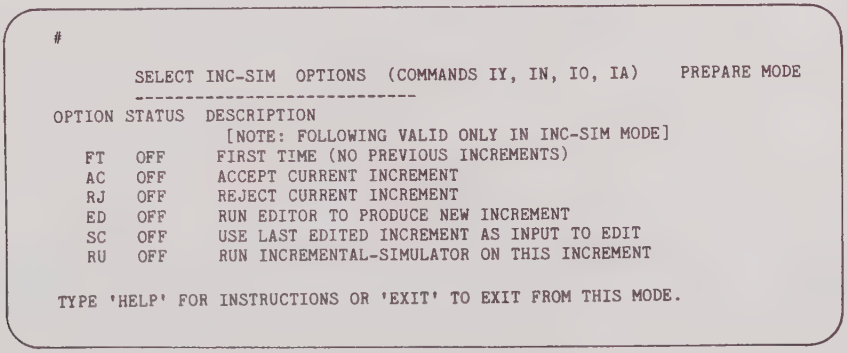
SF (SELECTED FAULT CLASSES MODELLED FOR DETECTION SIMULATION). This option displays the .SP file following a Selected Fault Detection run (using TRYSF macro, see para B.2.4.2). When using this option, F and D options should be OFF.

B.2.3.4 FAULT MODELING page

Options O, L, H, S, B, and I specify the types of faults to be modeled by Incremental AFD. These options only pertain to the first time a set of faulting runs is started on the board. The types of faults must be specified before running TRYF. If modeled faults for a board are changed, TRYF must be run again.

B.2.3.5 SELECT INC-SIM OPTIONS page

The major options that control the operation of Incremental Simulation are listed in Figure B-2. These options are located on the SELECT INC-SIM OPTIONS page of PREPARE and are defined in the following paragraphs. For clarity they are described in a different order than they appear on the Monitor page shown in Figure B-2.



```
#

        SELECT INC-SIM  OPTIONS  (COMMANDS IY, IN, IO, IA)    PREPARE MODE
        ----------------------------
OPTION STATUS  DESCRIPTION
                 [NOTE: FOLLOWING VALID ONLY IN INC-SIM MODE]
   FT   OFF     FIRST TIME (NO PREVIOUS INCREMENTS)
   AC   OFF     ACCEPT CURRENT INCREMENT
   RJ   OFF     REJECT CURRENT INCREMENT
   ED   OFF     RUN EDITOR TO PRODUCE NEW INCREMENT
   SC   OFF     USE LAST EDITED INCREMENT AS INPUT TO EDIT
   RU   OFF     RUN INCREMENTAL-SIMULATOR ON THIS INCREMENT

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure B-2. SELECT INC-SIM OPTIONS Page.

RU (RUN MACRO, CTRANS, AGN, EVAL, INVERT ON THE CURRENT INCREMENT) This option, when specified, causes the APG modules to perform simulation on the current increment. This creates the various data bases (.ET, .IN, .GO, .GV or their equivalent) needed to perform simulation on the current increment. As previously mentioned, there are two sets of files, the "accepted" set and the "trial" set. Thus, the actual names of the files created during the RU Sub-Mode depend upon whether FT is ON or OFF. If FT is ON, then the "accepted" set is created, otherwise a "trial" set is created.

FT (FIRST TIME NO PREVIOUS INCREMENT) Specifies that a new or initial set of "accepted" files is to be created based on the current N= and T= specifications.

NOTE

> When FT is OFF, the .SC file on working device 1 is the input test program for I Sub-Mode.

AC (ACCEPT LAST INCREMENT) If this option is on, the "trial" set of files, that was created the last time RU was ON, wil be appended to the "accepted" set of files. The file specified by T= will be updated with the new .TP.

RJ (REJECT LAST INCREMENT This option is used to reject the "trial" set. Primarily this option is used at the end of the test program generation to delete all unnecessary files associated with the "trial" segment. It is also used to clean up the disk if a different test program/board is to be processed. It is important not to leave unaccepted files of a board on the working disks while processing other boards.

NOTES

> Whenever FT is specified, a new "accepted" set of files is created. Thus any previously existing "accepted" set of files of the board is deleted.
>
> Since AC adds a "trial" set of files to the "accepted" set of files, a run with FT ON and at least one run with FT OFF must have been run before AC can do anything.
>
> If both AC and RJ are ON the AC operation will occur.

ED (RUN EDIT TO PRODUCE NEW INCREMENT) This option allows incremental simulation operation to automatically enter the editor, allow the specified segment to be edited, and return to the current incremental simulation operation.

SC (USE LAST EDITED INCREMENT AS INPUT TO EDIT) This option specifies that in the absence of a .SC the user wishes to use the .TP as the input to the editor. This option provides the capability to automatically specify the first test program increment or the accepted test program as the program to be edited when creating the next increment.

B.2.4 Using the Incremental Simulation Sub-Mode of PREPARE

Before using the Incremental Simulation Sub-Mode of PREPARE (see Figure B-3) the user must have generated the .SR file and run it through the N Sub-Mode of PREPARE.

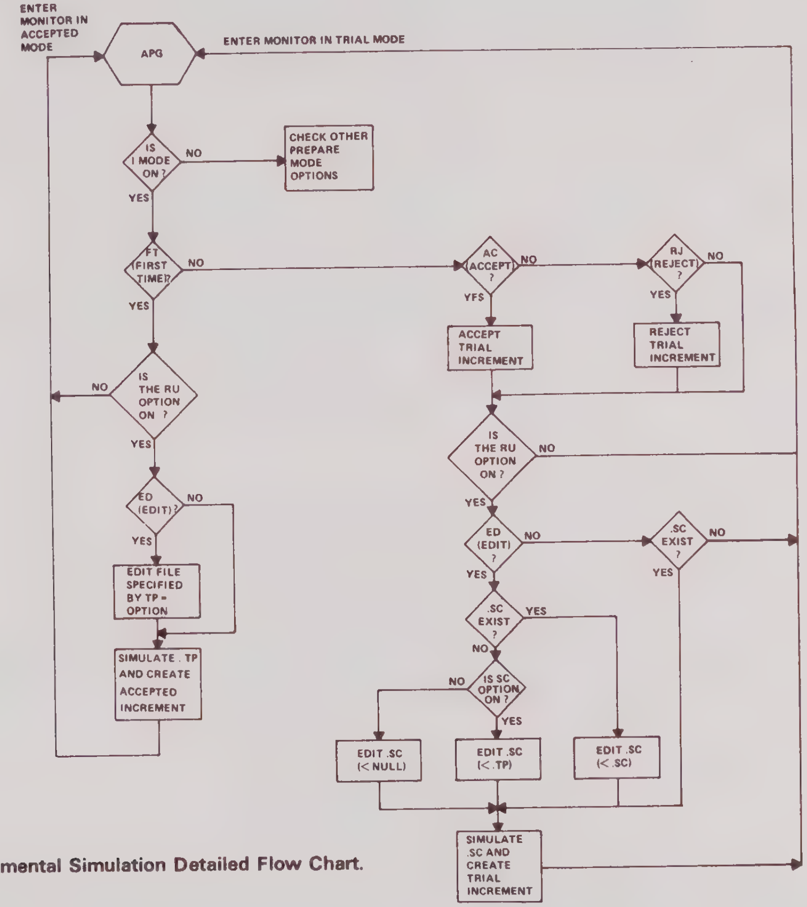

ENTER MONITOR IN ACCEPTED MODE
APG
ENTER MONITOR IN TRIAL MODE
IS I MODE ON ?
NO
CHECK OTHER PREPARE MODE OPTIONS
YES
FT (FIRST TIME)?
NO
YES
AC (ACCEPT) ?
NO
RJ (REJECT) ?
NO
YES
YES
ACCEPT TRIAL INCREMENT
REJECT TRIAL INCREMENT
IS THE RU OPTION ON ?
NO
YES
IS THE RU OPTION ON ?
NO
YES
ED (EDIT) ?
NO
YES
EDIT FILE SPECIFIED BY TP = OPTION
SIMULATE .TP AND CREATE ACCEPTED INCREMENT
ED (EDIT) ?
NO
YES
.SC EXIST ?
NO
YES
.SC EXIST ?
YES
NO
IS SC OPTION ON ?
NO
YES
EDIT .SC (< NULL)
EDIT .SC (< .TP)
EDIT .SC (< .SC)
SIMULATE .SC AND CREATE TRIAL INCREMENT

Figure B-3. Incremental Simulation Detailed Flow Chart.

B.2.4.1 First Time Operation

Once the .SR file is generated and the N Sub-Mode is run, the user accesses the PREPARE Mode, specifies Incremental Simulation, First Time operation, enables the editor, specifies that the previously generated test program segment is to be edited (if it exists and is desired) and specifies that incremental simulation is to be run. This can be done from the Monitor top level by using the FT batch job:

```
"FT
```

or by using the following command line:

```
PRE;EO I;IO FT,ED,RU;RUN;
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation
IO FT,	specifies First Time operation
ED,	enables call to editor on errors
RU;	run incremental simulation on this segment
RUN	executes the command line

NOTE

Do not include FINISH CAPS or END statements in the initial segment or "trial" segments.

Now the user either immediately begins simulation or edits the existing test program segment. Once this FIRST TIME segment has been edited and the user is satisfied with the segment, he exits the editor and the segment is run through MACRO (see para 5.2.3.1), CTRANS (see para 5.2.3.2), AGN (see para 5.2.4), EVAL which checks both the LOOP and CASE statements to determine if the conditions stated in these statements are present (see para 5.8.5) and INVERT (which produces the .GV file).

After the segment is processed the user may run the utility program DISPLAY to view the test program just produced. If the segment is satisfactory the user may write additional segments as necessary (see para B.2.4.2). If the segment does not accomplish the desired results, E/T/C Mode may be entered and either generate equations for the nodes in question or track activity back to its source. The user can then perform another first time operation which destroys the previous segment and generates a new first time (i.e. accepted) segment.

TRYF batch job

This batch job is used to examine the Fault Detection provided by the initial segment created by FT. Several options of PREPARE Mode must be specified before using TRYF. Turn ON Print Listings options F and D to display the .FP and .FG files, turn ON desired faults under Fault Modeling, and turn ON Select Miscellaneous Options C and V (or I and U, if desired).

This batch job is executed from the Monitor top level by entering

```
"TRYF
```

or by issuing the following command line:

```
PRE;EO F;SF=N;PP=N;RUN
```

where:

PRE;	specifies PREPARE Mode
EO F;	specifies Fault Detection simulation
SF=N;	detects all faults (NO Selected Faulting)
PP=N;	processes entire file (NO Partial Processing)
RUN	executes command line

B.2.4.2 Writing Additional Segments

Additional segments may be added to the test program at any time with the use of the TRYNW, TRY, and TRYNE batch jobs described in the following paragraphs. User-styled batch jobs defined in the initial or previously accepted sections can be called for use in trial segments.

TRYNW batch job

This batch job can be used to:

1. Reject the current increment,
2. Use the Editor to produce a new increment,
3. Run Incremental Simulation on the new increment.

The batch job is executed from the Monitor top level by issuing a

```
"TRYNW
```

command, or by issuing the following command line:

```
PRE;EO I;IO ED,RJ,RU;RUN
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation Sub-Mode
IO ED,	calls the Editor to produce a new increment
RJ,	rejects the current increment
RU;	runs Incremental Simulation on this increment
RUN	executes the command line

The first time this batch job is used, the editor comes up blank, ready for a new "trial" segment to be written. As with any batch job that invokes the Edit Mode, the Edit X and XMIT command must be executed from the editor once the new increment has been prepared in order for the next batch job segment (incremental simulation) to be executed.

TRY batch job

The TRY batch job provides the ability to continually modify and retry a new test-program segment until the user is satisfied with the resulting segment. When using the TRY batch job, the editor does not come up blank the first time, as it does with TRYNW. The editor will contain the last "trial" segment or the initial section if no "trial" segment has been written. Therefore, TRY can also be used in place of the TRYNW batch job if the user desires to use part of the initial section (if used immediately after FT) again.

To TRY a test-program segment, simply enter the following batch job from the Monitor top level:

```
"TRY
```

or enter the following command line:

```
PRE;EO I;IO ED,SC,RJ,RU;PP=N;RUN
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation
IO ED,	enables call to editor on errors
SC,	use previously generated test-program segment for editing purposes if an .SC does not exist
RJ,	delete previous "trial" file and provide more efficient disk usage
RU;	run incremental simulator on this segment
PP=N;	do not use partial processing results
RUN;	Execute the command line

The system automatically enters the editor, and if a previous test program segment (.SC file) exists it is displayed on the TEC screen. If a previous test program does not exist, then the .TP file is used.

On exiting the editor through use of the X and XMIT commands the "trial" segment is processed by the Incremental Simulator.

The utility program DISPLAY is used to view the test program segment just produced. If the segment is satisfactory the user accepts it (see para B.2.4.3) and then writes additional segments as necessary. If the segment does not accomplish the desired results, the user just performs another TRY operation and the previous "trial" segment is destroyed and a new "trial" segment is generated.

NOTE

> The reject option in the TRY batch job deletes all unnecessary files before attempting a new TRY.

TRYNE batch job

This batch job can be used to run Incremental Simulation on the current existing increment. This can be useful, for example, after correcting a problem in the .SR file and rerunning N Sub-Mode of PREPARE and FT. The last trial segment can be tested again without having to enter and leave the editor by using TRYNE. The batch job is executed from the Monitor top level by issuing a

```
"TRYNE
```

command, or by issuing the following command line:

```
PRE;EO I;IO RU;PP=N;RUN
```

where:

PRE;	enters PREPARE Mode
EO I;	turn the Incremental Simulation Sub-Mode ON
IO RU;	runs Incremental Simulation on this increment only
PP=N;	processes entire file (NO Partial Processing)
RUN	executes the command line

TRYSF batch job

This batch job determines and displays the Fault Detection provided by the test program including all "accepted" segments and the current "trial" segment. The Detected Faults List (.FP file) displayed includes the results of the "trial" segment. The Fault Group Detection (.FG file) does not include the results of the "trial" section. This file will be updated only after the section is "accepted". Several PREPARE Mode options must be specified before using TRYSF. Turn ON Select Miscellaneous Options C and V (also U and I, if desired). Turn ON either Print Listings options F and D, or SF (not all three). If F and D are used, the listing displayed following a Fault Detection run will contain all faults, whether detected or undetected. If option SF is used, only previously undetected faults will be displayed.

To run the batch job from the Monitor top level type the command

```
"TRYSF
```

or enter the command line:

```
PRE;EO F;SF=Y;ED=Y;PP=N;RUN
```

where:

PRE; specifies PREPARE Mode EO F; specifies Fault Detection simulation SF=Y; detects Selected Faults ED=Y; calls the editor to enter Selected Faults PP=N; do not use partial results from previous runs RUN executes the command line

When the command is executed the editor is called. Enter the Selected Faults to simulate (see para 5.2.6.3) or ALL UNDETECTED;. Exit the editor with X and XMIT.

PP batch job

This batch job creates the .FG and .FP files resulting from the test program and all "accepted" segments. Before using this batch job, turn ON Print Listings options F and D, and turn OFF option SF. The .FG and .FP files are automatically deleted by TRYNW, TRY, AC, and RJ. Use this batch job to view Fault Detection of the test program after using TRYNW, TRY, AC, or RJ.

To run the batch job from the Monitor top level enter

```
"PP
```

or enter the following command line:

```
PRE;EO F;SF=N;PP=Y;RUN
```

where:

PRE;	specifies PREPARE Mode
EO F;	specifies Fault Detection simulation
SF=N;	detects all faults (NO Selected Faulting)
PP=Y;	include processing results from previous runs
RUN	executes command line

TRYD batch job

This batch job runs the D Sub-Mode of PREPARE to produce a new binary data base for use in the DIAGNOSE Mode. The batch job is executed from the Monitor top level by issuing a

```
"TRYD
```

command, or by issuing the following command line:

```
PRE;EO D;RUN
```

where:

PRE;	enters PREPARE Mode
EO D;	specifies D Sub-Mode of PREPARE
RUN	executes the command line

B.2.4.3 Accept Procedures

When the new TRY segment is satisfactory (see para B.2.4.2) it is appended to the existing test program by performing an ACCEPT operation. When a segment is appended, it becomes a part of the test program and can not be altered unless a complete new first time run is performed. To accept a segment enter the following batch job:

```
"AC
```

or enter the following command line:

```
PRE;EO I;IO AC;RUN
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation
IO AC;	accept this increment
RUN	executes command line

B.2.4.4 Reject Procedures

The reject procedure is intended to be used only when the current TRY segment is of no value. This procedure will delete all trial files associated with the last test-program segment generated. There are two reasons for using the "reject" procedures. First, such files consume valuable space and second, there is an interaction between the PREPARE Mode and the E/T/C and DISPLAY Modes, such that if under certain conditions "trial" files exist, these display modes assume the user is interested in reviewing the trial files.

It is not necessary to REJECT a test-program segment each time it is to be retried, since each new TRY automatically deletes the previous "trial" set before it proceeds. However, it is good practice to run RJ with each trial simulation to promote efficient disk utilization.

To perform a reject operation enter the following batch job:

```
"RJ
```

or enter the following command line:

```
PRE;EO I;IO RJ;RUN
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation
IO RJ;	reject this increment
RUN	executes the command line

NOTE

> RJ is included in the TRY batch job to eliminate the necessity of separately rejecting files with each simulation.

B.2.5 Using the I Sub-Mode With the Display Mode

There are two ways of reviewing the test program with the DISPLAY Mode. If only the ACCEPTed part of the test program is to be displayed the user must enter PREPARE Mode, enable Incremental Simulation, and specify First Time operation (do not specify AC, RJ, or RU), and RUN it. This may be done by entering the following batch job:

```
"SETA
```

or by entering the following command line:

```
PRE;EO I;IO FT;RUN
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation
IO FT;	specifies first time operation
RUN	executes command line

If the complete test program ("accepted" and "trial" segment) is to be displayed the user must enter PREPARE Mode and run Incremental Simulation with all the SELECT INC-SIM Options OFF. This may be done with the following batch job:

```
"SETT
```

or by entering the command line:

```
PRE;EO I;IO;RUN
```

where:

PRE;	specifies PREPARE Mode
EO I;	specifies Incremental Simulation
IO;	specifies all INC-SIM options are OFF
RUN	executes the command line

Next, the DISPLAY Mode can be entered and used.

NOTE

> Figure B-3 indicated which files are automatically displayed for review by the note associated with the return path to the APG Monitor.

B.2.5.1 Incremental Simulation File Usage

The initial test program segment is supplied to the Incremental Simulator in the file specified by the T= option in the File Usage Page. Subsequent incremental segments (FT=OFF) are passed to the Incremental Simulator in the .SC file on working device 1. As these incremental segments are "accepted", they are accumulated in the .YC file on working device 1 and also in the file specified by the T= option.

If it is necessary to make a change in the already "accepted" portion of the test program, turn on the FT, ED, and RU options. FT specifies that the T= file (which contains the original initial segment plus all the "accepted" increments) is to be used as input for the new initial segment. ED specifies editing and RU initiates the execution.

NOTE

> Do not edit the .TP file without rerunning First Time.

B.2.5.2 Incremental Diagnosis

During program development, the recommended procedure is to verify sections of the test program as they are created. This verification can be performed using Incremental Diagnosis.

The initial portion of the test program can be verified by using the TRYD batch job or running the D Sub-Mode of PREPARE to create the data base files, and then entering and running the DIAGNOSE Mode to greenlight a UUT. Once the test program initialization has been verified, additional sections can be created by Incremental Simulation, verified in DIAGNOSE Mode, and then accepted and added to the test program.

Incremental Diagnosis allows the user to verify the Incremental Simulation "trial" sections prior to their acceptance. The procedure is to create the trial sections using the TRYNW and TRY batch jobs, and then use TRYD which runs the D Sub-Mode of PREPARE . This action will build the .DA file using both the previously accepted portions of the test program and the unaccepted section. The user then runs the DIAGNOSE Mode to test the UUT and verify the accuracy of the added section. If the new section passes, the user then leaves DIAGNOSE Mode and accepts the trial section. If the new section fails, the user leaves DIAGNOSE Mode, rejects the trial section, and then creates a new "improved" trial section.

B.2.6 Test Generation Language

The Test Generation Language (TGL) represents a new approach to the test plan implementation. It provides two new features: internal and external board state confirmation, and selection of goal-directed test sequences. Given a statement of the user's goal, the system will automatically determine whether or not that condition is met. The Test Generation Language can be used as a general tool to document test strategies, to verify that test goals are satisfied, or to select the number of iterations of a repetitive sequence required to achieve a particular goal with a REPEAT n UNTIL statement. The last is useful for generating clocking sequences. If several strategies seem likely to achieve a specified state, each may be included as a potential sequence in a CASE statement. The system will select the first sequence that succeeds.

B.2.6.1 REPEAT n UNTIL (condition)

This command allows the user to specify a test sequence that is to be appended to the test program n times (maximum) or as many times as necessary to meet the specified condition.

The .REPEAT n UNTIL command format is as follows:

```
.REPEAT n UNTIL (condition);


 test sequence


.LOOP;
```

When the .REPEAT n UNTIL command is encountered by the software, the software first verifies that the specified condition has not been met. If the condition has been met then the software skips over the .REPEAT n UNTIL test program segment and continues on with the remaining test program. If the condition has not been met, then the software begins processing the test sequence commands following the .REPEAT n UNTIL command. The condition is checked each time a .LOOP statement is encountered. When the condition is true then the software skips the remaining number of iterations and continues on with the program.

This construct has two significant applications. The first is to insert the appropriate number of iterations of a repetitive sequence, for example a clocking sequence, without concern with the exact number of iterations required. Note, however, that the condition evaluation process requires running the incremental simulator on each iteration. It should not be performed unnecessarily. If, for instance, it is clear that at least n iterations of a sequence are required a separate non-conditional REPEAT n loop should be used to precede the conditional REPEAT n UNTIL loop.

A second, less obvious, application is to test whether a specified internal or external condition is met. This application takes advantage of the .STOPLOOP command (see Para. B.2.2.2) which causes the simulation to be suspended if a specified condition is not met. For this application, a REPEAT n UNTIL loop with a null sequence (or a dummy test sequence) is specified.

```
EXAMPLE:      .REPEAT 1 UNTIL (IC3.5=1);

              .STOPLOOP;
```

This example causes simulation to be suspended and a message that the condition is not met to be issued if IC3.5 is not set to 1 by the earlier test sequences. Otherwise any later sequence will be simulated.

The variables used in the REPEAT command are n and condition. The n variable is a positive, non-zero integer and is the maximum number of times the test sequence is to be repeated. The condition specification variable is a Boolean expression specifying the looping termination condition.

The Boolean expression for the condition is composed of the terms of the form:

```
ICNAME.P=V

   or

EXTERNAL=V
```

The value specification, V, may be expressed with any of the following values:

X	Specifies indeterminate state
N	Specifies negative pulse
P	Specifies positive pulse
R	Specifies race condition
0	Specifies logic 0
1	Specifies logic one

ICNAME is the name of the IC and P is the pin number associated with that IC. External is any valid external name.

If the states of more than one node are to be specified as a condition, references to these states may be combined using the Boolean operators: AND, OR, and NOT. The AND operator is expressed with an ampersand (&), the OR operator is expressed with an exclamation mark (!), the NOT operator is expressed with a minus sign (-).

Terms of the boolean expression for the condition are evaluated subject to the following rules:

NOT(-) Operations are performed first

AND(&) Operations are performed second

OR(!) Operations are performed last

Parentheses may be used to override the specified order of operation in an expression. When sets of parentheses appear the terms within the parentheses are evaluated first. When sets of parentheses appear within other sets of parentheses, the expression within the innermost set is evaluated first, then the expression in the next set and so on.

An example of the .REPEAT n UNTIL command is:

```
.REPEAT 25 UNTIL (IC7.2=1&(IC2.1=0!-IC3.5=X));
```

This command specifies that the program segment will be repeated 25 times or until

both IC7.2 is a logic one

AND

either or both of the following conditions are met;

IC2.1 is a logic zero

OR

IC3.5 is NOT in an indeterminate or X state.

B.2.6.2 .LOOP/.STOPLOOP Termination

There are two types of terminations for a .REPEAT n UNTIL command. The first is with the .LOOP statement which indicates that if after the specified number of loops (n) the condition specified has not been met, the message:

```
TGL CONDITION NOT MET
```

should be displayed and the software should continue on with the program. The second type of termination is the .STOPLOOP command which after the specified number of loops (n) are exhausted and the conditions specified are still not met, the .REPEAT n UNTIL operation should terminate and the following error message be displayed:

```
REPEAT-STOPLOOP FAILED TO PRODUCE UNTIL'S CONDITIONS
```

B.2.6.3 CASE UNTIL

The CASE command allows specification of alternative test sequences, each of which will potentially satisfy the specified condition. Each of the specified sequences are successively "tried" until the condition is met. Successful sequences are in effect "accepted" by the system and appended to the "trial" test program. Unsuccessful sequences are automatically rejected by the system.

The case command format is as follows:

```
.CASE UNTIL (condition);

  sequence 1
                          \
.MORECASE;                 |
                           |  There is no limitation to the
  sequence 2               |  number of sequences.
                           |
.MORECASE;                /

  sequence 3

.ENDCASE;
```

The form and processing of the condition specification is identical to that in the REPEAT n UNTIL command (see para. B.2.6.1). If a test sequence does not meet the conditions specified then that segment is not included in the resultant specified test program and the next segment is processed. If a segment produces the condition then the remaining segments are not processed or included in the resultant test program. Only the segment that produces the condition is included in the resultant test program. Note that when the system encounters a CASE statement, the simulated board state is checked to see if the condition is already satisfied. If so, none of the cases specified will be "tried".

The following is an example of a CASE command:

```
             .CASE UNTIL (IC2.3=X&IC3.2=X!IC4.3=P);

             {  IL(76) IH(5);
  Test       {
  Sequence   {  IH(4) IL(3);
     1
                .MORECASE ;

  Test       {  IH(24) IL(16);
  Sequence   {
     2          .MORECASE ;

  Test       {  IH(42) IL(21);
  Sequence   {
     3          ENDCASE;
```

This command specifies that each program segment will be successively tried and the condition checked before each attempt. If the segment does not cause the condition be true the segment is not included in the resultant test program and the next segment is processed. This continues until the condition is true or an .ENDCASE or .STOPCASE statement is encountered.

The condition in this example states that each additional segment will be tried until:

IC4.3 exhibits a positive pulse

OR

both IC2.3 is in an indeterminate state

AND

IC3.2 exhibits an indeterminate state.

There are two types of termination for the CASE command. The first is the ENDCASE statement (signifying that no case selected met the condition) which terminates the CASE operation, displays the message

TGL CONDITION NOT MET,

and allows the software to continue to process the remaining statements of the program. The second is the STOPCASE statement which terminates the CASE and terminates simulation on this increment if the condition was not met. In this instance the following error message is printed:

CASE-STOPCASE FAILED TO PRODUCE UNTIL'S CONDITION

NOTE

> The processing of the TGL statements REPEAT n UNTIL and CASE UNTIL requires the use of the Incremental Simulator. Thus these statements are valid only in the Incremental Simulation Sub-Mode of PREPARE and in no other CAPS Mode or Sub-Mode.

B.2.6.4 Important Points Concerning TGL Statements

If TGL statements are used, they will appear in the .TP file in their original form and in the .ET file only in the expanded form. The .ET file contains only the statements that are required to achieve the specified condition and no remnants of the original TGL statements. Thus the .ET file is a valid test program for the basic CAPS simulation system.

Due to the overhead associated with the TGL system, it is wise to save both the .TP which contains the original test generation language constructs and the resultant .ET file or .DA file as well as the corresponding .SR file.

If a board is considerably reworked and a new test program must be regenerated, using the .TP file will enable the original strategies and condition checking to be processed and may simplify the task of updating the test program. If the board changes are minor, using the .ET or .DA file may be a worthwhile shortcut although it forfeits the opportunity to automatically verify that conditions are met as needed and that appropriate CASE and LOOP interactions are generated (see para. B.2.6).

The ENDCASE and LOOP statements generate a Teletype warning of the form:

TGL CONDITION NOT MET

This is just a warning, however, and test program processing will continue.

B.3 E/T/C MODE

B.3.1 General

Equation Generation, Tracking, and Connectivity (E/T/C)Mode provides a fast and efficient means of tracking simulator related problems (values that are not propagated, inactive nodes, indeterminate (or "X") states, sources of pulses, and level transitions) to their source. It also can be used to obtain a better understanding of the circuit being simulated so that necessary improvements can be added to the test program used to drive the simulated model.

Equation Generation and Connectivity Analysis are primarily tools that can be used to help a user develop an efficient test program by allowing the software to analyze complex circuit segments. It might be necessary, for example, to determine those inputs and paths that actually affect an active circuit pin (or node) in a complex circuit and/or determine the actual input states that are required to drive a pin (or node) into a particular high or low state. Armed with this information, the user can rapidly develop effective test strategies and incorporate them into the test program. Both Equation Generation and Connectivity Analysis may be used after the Network Description (.SR file) has been prepared and run through the N Sub-mode of PREPARE.

Activity Tracking is a tool that can be used to "debug" completed test programs. It is used in conjunction with the Good Board Simulation Results (PRETTY) output, and can help pinpoint the causes for such problems as X (or indeterminate states), positive or negative going pulses, and level changes (Transitions) on active pins occurring at any test step. Activity Tracking may be used after the Network Description (.SR file), and basic test program (.TP file) have been prepared and run through the N, T, and G sub-modes of PREPARE. It may also be used after running the N and I sub-modes of PREPARE. Each of the tools that are available for use in the E/T/C Mode are described further in the following paragraphs.

B.3.1.1 Back Connectivity

Back Connectivity is essentially a sub-set of activity tracking that can be used to determine the input pins and circuit paths that are affected by an active pin or node within a circuit. Knowledge of circuit connectivity can provide insight into circuit interactions and is thus useful in partitioning a circuit into independent sections. Basically, back connectivity can show how complex logic is interconnected by listing both paths taken and inputs to any pin or node within the circuit. This type of information is useful in determining where logical circuit partitions occur or for rapidly acquiring the information necessary to make faults at these pins or nodes detectable. It may also be used to determine how circuit library models are actually connected without reference to the .SR or .XR files in order to analyze connections. Modeling peculiarities may therefore be tracked down "on-line" through the use of Back Connectivity.

B.3.1.2 Equation Generation

Equation Generation provides Boolean expressions for signals in the circuit being tracked. These equations are displayed using an easily understood format to specify the relationship of a given node to other relevant nodes of the circuit. Such equations may be useful for test generation purposes since they highlight the enabling conditions of a circuit. Equations can be listed that are necessary to drive a particular pin or node into either a high or low state, thus making it possible to rapidly gain knowledge of the circuit and improve the test program.

B.3.1.3 Activity Tracking

Activity Tracking allows the tracking of an active event to its source. Tracking quickly pinpoints causes for the following types of problems that may have been noted as the result of an analysis of the Good Board Simulation Results (PRETTY) printout for the board being simulated:

1. X (or indeterminate) state. These states may be caused by critical race conditions, oscillations, improper initialization, etc. Activity tracking enables X states to be tracked back to their source from deterministic states at any test step.

2. Positive or Negative going pulses occurring at any test step.

3. Positive of Negative going transitions (level changes) and pulses occurring at any test step.

Each of these events may be tracked manually (level-by-level) of logic through a circuit or automatically (without stopping at intermediate levels) back to the source inputs. Tracking may also be terminated whenever a memory, CAPS MSI device (a recursive CAPS/APG library element), or specified circuit level (depth) is encountered. Displays are flexible and can be formatted to show all paths taken or just the final active levels at the tracking termination points so that only relevant (useful) data need be generated.

B.3.2 E/T/C FILE USAGE Page

The E/T/C Mode is entered from the APG Monitor only (not the CAPS Monitor) using any of the procedures listed in Section 3. For example,

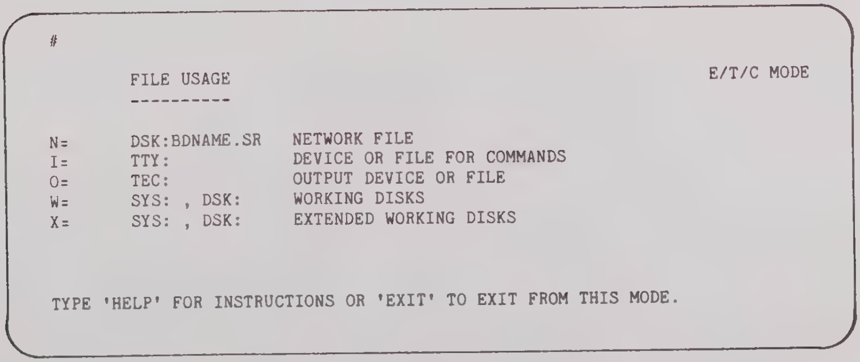
```
ETC and XMIT
```

will cause the E/T/C Mode to be entered and the FILE USAGE Page shown in Figure B-4 to be displayed. Options on this page perform the following functions:

1. The Network File Name (N=)

   This is the name and location of the NETWORK FILE that has already been run through the PREPARE Mode. For example,

   ```
   N=DSK3:SAMPLE.SR
   ```



```
#

        FILE USAGE                                           E/T/C MODE
        ----------

N=      DSK:BDNAME.SR    NETWORK FILE
I=      TTY:             DEVICE OR FILE FOR COMMANDS
O=      TEC:             OUTPUT DEVICE OR FILE
W=      SYS: , DSK:      WORKING DISKS
X=      SYS: , DSK:      EXTENDED WORKING DISKS


TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure B-4. E/T/C Mode FILE USAGE Page.

specifies the network description file SAMPLE.SR on DSK3:. If a device and file name extension are not specified, DSK1: and the .SR extension are assumed.

2. Device or File for Commands (I=).

This is the device or device and file name for entry of E/T/C commands. Commands that enable the E/T/C Mode to perform activity tracking, back connectivity, and equation generation may be issued from the device or device and file specified by the I= option. For example, if,

```
I=TTY:        or        I=TEC:
```

then commands may be issued, and changed from either the Teletype or Display Terminal Keyboards (depending on which one was specified). This is the procedure that would normally be followed as it enables the user to issue and/or change commands "on-line" at any time. Commands may also be placed in a source file that is prepared in advance by the EDIT Mode and saved on a file structured disk device. If the device and file are then specified as in the following example,

```
I=DSK2:NAMEA.AB
```

the command input file will be called and executed. This procedure can be accomplished as part of a BATCH run, if desired (refer to Para. 3.3.7). If a device is not specified, DSK: is assumed. It is therefore only necessary to specify the file name and extension (any unused extension may be listed for files located on DSK: as the following example indicates:

```
I=NAMEA.AB
```

3. Output Device or File (O=)

This is the output device or device and file that the results of an E/T/C Mode run are output to. If TEC: is specified, for example,

```
O=TEC:
```

all displays resulting from an E/T/C Mode run will be output to the display terminal. If hard copies of the displays are required, then the line printer or Teletype

```
O=LPT:        or        O=TTY:
```

may be specified. If the output of an E/T/C run is to be saved (perhaps for examination at a later time because a BATCH run is in progress) a device and file may be specified:

```
O=DSK3:NAMEB.LS
```

The results of the E/T/C Mode run will then be output to DSK3:NAMEB.LS and can be retrieved at a later time for detailed analysis. If a device and file extension are not specified, the DSK: device and .LS extension are assumed so that it is only necessary to specify

```
O=NAMEB
```

if the file is to be located on DSK: and use the .LS extension. Note that files *cannot* be saved on the *first* working disk specified by the W= option. Any other disk devices may be listed, however.

4. *Working Disks (W=)*.

Enter the *same* working devices that were specified during the PREPARE Mode run for the board listed. CAPS/APG .ST,.NM, AND .GV files are required in order to run activity tracking, while the .ST and .NM files are required for back connectivity and equation generation. The devices that contain these files *must* be specified at this time. For example, if DSK2:, DSK3: were specified for the original PREPARE Mode run, then these same devices would contain the needed CAPS/APG files for an E/T/C Mode run. Under such conditions, the W= option would be listed as:

W=DSK2:, DSK3:

B.3.3 E/T/C Command Mode

Once the E/T/C Mode has been entered and the FILE USAGE Page options set-up, the E/T/C Command Mode can be entered by issuing the command

RUN and XMIT

E/T/C Commands may be issued from the Teletype keyboard (if I=TTY:), the display terminal keyboard (if I=TEC:), or from a source file (if I=DSKn: NAME). Display outputs resulting from the run are output to the device named by the O= Option. The E/T/C Mode itself will operate in a slightly different manner depending on the combination of Input/Output devices selected.

If TEC: has been specified as the Output device, and TTY: as the device for commands, the ETC OPTION STATUS Message will be displayed at this time. This message lists the current status of optional commands and is described further in Para. B.3.3.3. The Teletype also responds by printing a prompt (>) character to signify that it is ready to accept an E/T/C command. Teletype commands are executed by entering the command after the prompt, and then depressing the RETURN (↲ key to execute the command.

If the display terminal keyboard has been specified as the device for commands (I=TEC:), the prompt (>) character does *not* occur. Commands may be entered directly at the cursor *home* position, and executed by depressing the XMIT key. Commands can only be issued *if* the display terminal has *first* been cleared of any data displays by depressing the XMIT key.

If a source file has been specified as the device for commands (I= DSKn:NAME) the commands in the file will be executed automatically when the RUN command is issued.

The following discussion assumes that TEC: *has* been specified as the Output device, and TTY: as the device for commands. Table B-1 lists the complete command set that may be specified. Basically, the command set consists of the following major segments:

1. Four Format Display Control Commands (FORMAT Commands) that control the amount and type of information to be output in displays. These commands enable a user to select a large amount of data or only the relevant in-

Table B-1

E/T/C MODE COMMAND SET

Command	Use
FORMAT DISPLAY CONTROL COMMANDS	
FORMAT UPDATE ALL or ACTIVE	For Updated Display only. Lists ALL input pins at that point (whether active or not) or the ACTIVE pins only.
FORMAT PATH ALL or ENDS	For Updated Display only. Lists ALL active paths back to the source pin, or only the active ENDing paths.
FORMAT FINAL COMPLETE or SUMMARY	COMPLETE lists all active paths, and all termination points (ends). SUMMARY lists final termination points (ends) only.
FORMAT EQUATION LIST or TABLE	Equations may be shown as a LIST of terms (long form) or as a TABLE (short form).
INITIATE PROCESS START COMMANDS	
TRACK p UNTIL MEMORY, MSI, LEVEL n, NONE	TRACK active pins or nodes.
BACK p FULL or SIMPLE UNTIL MEMORY, MSI, LEVEL n, NONE	Identify the circuit paths affected by an active pin or node.
EQUATION p ONE or ZERO UNTIL MEMORY, MSI, LEVEL n, NONE	List those circuit pin states that must be driven High or Low in order to set a specified pin High or Low.
GENERAL COMMANDS	
STEP n	TRACKing test step from PRETTY File.
UPDATE PIN p LEVEL, BRANCH, NONE	Halt the process at the point specified and output an updated display.
CONNECT	List final display.
EXIT	Close any open files and Exit to the APG Monitor.
TELETYPE COMMANDS	
CTRL C	Exits to APG Monitor directly. Does not close output files.
CTRL X	Exit back to E/T/C. Command Mode.
CTRL U	Terminate command line but remain in E/T/C Command Mode.
P	Copies the TEC screen display to the Line Printer, if active. If the Line Printer is off or not on the system, the TEC screen will be copied to the Teletype.
R	Change the level display switch from REPEAT to NEXT or vice versa. If in UPDATE LEVEL Mode and the level display switch is REPEAT the current level display will be repeated.

formation desired. FORMAT Commands must be issued first (ahead of Initiat Process Start Commands) as they determine the type of data to be output on the run has been started. Format Display control commands are described further in Para. B.3.3.1.

2. Three Initiate Process Start Commands (TRACK, BACK, and EQUATION that start the process indicated. These commands must always be entered last (after all other commands have been executed) as they start the process involved. Initiate Process Commands are described further in Para. B.3.3.2.

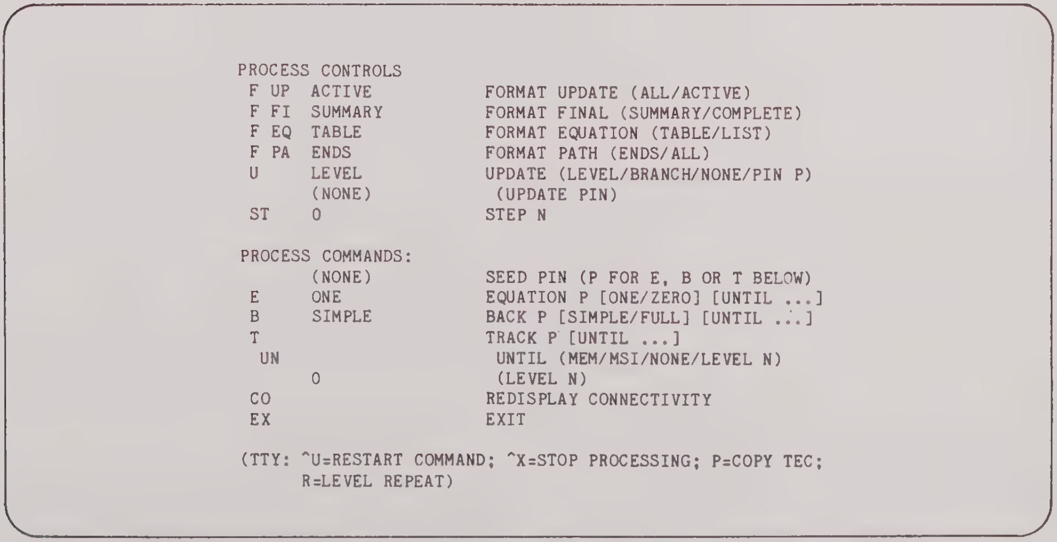
3. Four General Purpose Commands (STEP, UPDATE, CONNECT, and EXIT) that perform the various tasks described in Para. B.3.3.3(EXIT to the APG Monitor, UPDATE at a particular circuit point, etc.).

4. A special set of five commands (CTRL C, X, P, R, and U) that are always issued from the Teletype keyboard. These commands enable a user to terminate command lines, exit to the APG Monitor, etc., and are described in Table B-1.

Whenever the E/T/C command Mode is entered, the set of command options listed in the ETC OPTION STATUS Display (Figure B-5) are automatically assumed. As commands are issued, the status is automatically updated to reflect the current set of commands in use. A status display that reflects these commands is automatically displayed wher the system is ready for a new command. It is therefore only necessary to issue comman that change prior to issuing an Initiate Process Start Command. The complete set of commands that may be issued are described in the following paragraphs.

B.3.3.1 Format Display Control Commands

These commands establish the Output FORMAT of displays resulting from an E/T/C Mode run. The FORMAT UPDATE and FORMAT PATH Commands enable a user to select the type and amount of information to be included in updated displays that occur at selected points during a run. The FORMAT FINAL command selects the type and amount of information to be included in the final display that occurs when the Initiate Pro-



```
PROCESS CONTROLS
 F UP  ACTIVE            FORMAT UPDATE (ALL/ACTIVE)
 F FI  SUMMARY           FORMAT FINAL (SUMMARY/COMPLETE)
 F EQ  TABLE             FORMAT EQUATION (TABLE/LIST)
 F PA  ENDS              FORMAT PATH (ENDS/ALL)
 U     LEVEL             UPDATE (LEVEL/BRANCH/NONE/PIN P)
       (NONE)             (UPDATE PIN)
 ST    0                 STEP N

PROCESS COMMANDS:
       (NONE)            SEED PIN (P FOR E, B OR T BELOW)
 E     ONE               EQUATION P [ONE/ZERO] [UNTIL ...]
 B     SIMPLE            BACK P [SIMPLE/FULL] [UNTIL ...]
 T                       TRACK P [UNTIL ...]
  UN                      UNTIL (MEM/MSI/NONE/LEVEL N)
       0                  (LEVEL N)
 CO                      REDISPLAY CONNECTIVITY
 EX                      EXIT

(TTY: ^U=RESTART COMMAND; ^X=STOP PROCESSING; P=COPY TEC;
      R=LEVEL REPEAT)
```

Figure B-5. ETC OPTION STATUS Display

cess Start Command invoked (TRACK, BACK, or EQUATION) has finished the action indicated. The FORMAT EQUATION Command applies only to Equation Generation and enables a user to select either a list or a table format for equations generated during both updated and final displays. The command format for each command is as follows:

FORMAT UPDATE All or ACTIVE

This command applies to the updated display only. The updated display occurs only if the UPDATE command has also been invoked to cause the Initiate Process Start Command involved (TRACK, BACK, or EQUATION) to halt at specified circuit points and display the results at that IC pin, LEVEL, or BRANCH. One of the following options must also be included as part of the UPDATE command:

1. ALL. When ALL is specified, all Inputs at that stopping point (whether they are active circuit paths being followed or not) will be displayed.

2. ACTIVE. When ACTIVE is specified, only the active circuit paths being followed at that stopping point are listed.

FORMAT PATH ALL or ENDS

This command applies to the updated display only. One of the following options must be selected:

1. ALL. When ALL is specified, all active paths back to the source input are also listed.

2. ENDS. List the ending paths only.

FORMAT FINAL COMPLETE or SUMMARY

This command applies to the final display only. The final display occurs only after the Initiate Process Start Command involved (TRACK, BACK, or EQUATION) has finished the actions indicated. For example, after the TRACK command has completed all tracking specified. Either of the following options must be included as part of the FORMAT FINAL command:

1. COMPLETE. When COMPLETE is specified, the final display lists all active paths taken back to the termination of the action. Note that only the active paths taken are listed (not all possible circuit paths). In addition, all termination points are also included in the final display.

2. SUMMARY. When SUMMARY is specified, the final display lists only the final termination points. The paths taken are not included, thus reducing the total amount of data output.

FORMAT EQUATION	LIST or TABLE	This command applies to Equation Generation only. One of the following options must be specified: 1. LIST. When LIST is specified, both updated and final equations are included in a list of terms (long form). 2. TABLE. When TABLE is specified, both updated and final equations are included in a table of terms (short form).

B.3.3.2 Initiate Process Start Commands

These commands initiate (start) the process indicated (activity) TRACKing, BACK, or EQUATION) and also impose some of the limits or restrictions to be applied to the run. Because these commands start the process selected, they should always be issued last (after all other E/T/C Mode Commands have been issued.) The followin types of runs are possible:

TRACK	Once the N, T, and G sub-modes of PREPAR (or the N and I sub-modes) have been run the TRACK command can be used to track active pins or nodes that may be in an "X" state, have positive or negative going pulses, or have level changes (transitions) that occur at any test step in the PRETTY listing.
BACK	Once the N sub-mode of PREPARE has been run, the BACK command can be used to obtain paths and inputs affecting any pin or node. This data may prove helpful when generating the test program.
EQUATION	Once the N sub-mode of PREPARE has been run, the EQUATION command can be used to obtain those input or intermediate pin states that must be driven High or Low in order to drive a specified circuit pin High or Low. Such data can prove useful when generating the test program.

The TRACK Command has the following format:

```
TRACK p UNTIL MEMORY, MSI, LEVEL n, NONE
```

Where:

TRACK	Is the command to start the action indicated (TRACK). Only the underlined characters need be listed, however, all characters can be included for clarity, if desired.

p	Is the starting Point at which TRACKing is to begin. At least one space must be inserted prior to and after p. The character p may be one of the following: 1. An IC Input or Output pin. TRACK IC8.3 (Track IC8 pin 3) 2. An intermediate name. TRACK AE5 3. An EXTernal name. TRACK EXT8
UNTIL MEMORY, MSI, LEVEL n, NONE	The UNTIL construction limits the action being undertaken, if used. For example, TRACK IC8.2 UNTIL MEMORY will result in a termination of any active path being followed that includes a MEMORY element. Note, however, that only the path with a memory element will be terminated. All other active paths being followed will continue to be tracked until no active paths remain. The following terminators may be specified: 1. MEMORY. Terminate each path at the first circuit memory element encountered. 2. MSI. Terminate each path at the first element that has been entered into the CAPS/APG Library as a recursive element. 3. LEVEL n. Terminate the process when the LEVEL of circuit logic specified (n) is reached. A level of circuit logic is defined as a specific number of devices and ICs, as modeled, back from the starting point (p). For example, the command TRACK IC8.3 UNTIL LEVEL 5 would allow the software to track back from IC8.3 until five levels of logic (perhaps four devices and one IC) have been reached. Tracking for that particular active path would then terminate at that point. More than one delimiter may be specified by chaining them together with commas (,). For example, the command TRACK IC8.2 UNTIL MEMORY, MSI

would result in the action being terminated whenever the first MEMORY OR MSI device is encountered. Note that the level of circuit logic that is counted within an IC depends only on how that IC has been modeled in the CAPS/APG Library. Those IC elements that have been modeled as individual elements can be tracked.

4. NONE. If the UNTIL clause is not to be used as part of a command, but has been used previously, always include the

UNTIL NONE

construction once in order to erase any delimiters (MEMORY, MSI, LEVEL n) that may have been remembered from prior commands (this procedure is not required when the E/T/C Command Mode is first entered as NONE is assumed.

The BACK Command has the following format:

```
                     FULL
BACK    p                  UNTIL MEMORY, MSI, LEVEL n, NONE
                     SIMPLE
```

Where:

BACK	Is the command to start the action indicated (BACK)
p	Is the starting Point at which BACK is to begin. The character p may be any of the items listed under p for TRACK.
FULL or SIMPLE	When FULL is specified, all feedback loops are followed. When SIMPLE is specified, feedback loops are not followed.
UNTIL MEMORY, MSI, LEVEL n, NONE	The UNTIL construction limits the action being undertaken, if used. Methods of using the UNTIL clause are described under the TRACK command.

The EQUATION Command has the following format:

```
              ONE
EQUATION  p   or   UNTIL MEMORY, MSI, LEVEL n, NONE
              ZERO
```

Where:

EQUATION	Is the command to start the action indicated (EQUATION).

p	Is the starting Point at which EQUATION is to begin. The character p may be any of the items listed under p for TRACK
ONE or ZERO	Specifies whether the terms generated are to set the starting Point listed High (ONE) or Low (ZERO).
UNTIL MEMORY, MSI, LEVEL n, NONE	The UNTIL construction limits the action being undertaken, if used. Methods of using the UNTIL clause are described under the TRACK command.

B.3.3.3 General Purpose Commands

These commands serve a variety of purposes. They can be used for example, to specify the CAPS test step at which to begin tracking, stop and UPDATE at a particular pin, LEVEL, or BRANCH, display the current E/T/C OPTION STATUS, etc. The following General Purpose Commands may be issued:

STEP n	This command applies only to activity TRACKing, and is the means used to include test step numbers in E/T/C output listings. The command specifies the Good Board Simulation Results (PRETTY) test step at which activity occurred. It is not necessary to list the exact test step. The TRACKing software tracks back from the first test step prior to the one listed that has activity. If a test STEP number is not defined, then the most recently specified test STEP number issued is used when a TRACKing command is issued; otherwise, test STEP 0 is assumed. At least one space must be inserted between STEP and the number (n). While the STEP Command is only required for TRACKing, the test step number will also appear in BACK and EQUATION Output Listings that are generated.
UPDATE PIN p, LEVEL, BRANCH, NONE	This command halts the Initiate Process Start Command involved (TRACK, BACK EQUATION) at specified circuit points and outputs the updated display controlled by the various FORMAT commands. The UPDATE command allows a user to analyze actions occurring at circuit points and then either proceed or return to the command mode (CTRL X) where additional process control commands can be issued (perhaps to track other paths of interest). To proceed to the next point where another updated display will be output, a user can depress any Teletype key (space bar, Carriage Return, etc.). At least one of the following modifiers must also be included as part of the UPDATE command:

1. LEVEL. Stop at each circuit level. For example, if six devices and two ICs are involved, the process will stop eight times, and output the updated display at each stopping point (LEVEL). Note that this use of LEVEL is *not* followed by a number (n).

2. BRANCH. If BRANCH is specified, the process will stop and output an updated display whenever multiple active paths are encountered (whenever two or more active paths occur).

3. PIN p. If an IC PIN is specified, the process will stop and output the updated display *if* (and only if) the IC pin *listed is* encountered, however, the final display will be output. It is thus possible to determine if a path exists between the problem and a circuit point that a user thinks might be causing the problem. For example, the command

```
UPDATE PIN IC8.2
```

will cause the process to stop if (and only if) an active path that includes IC8 pin 2 is encountered. It is also possible to combine modifiers. For example, the command

```
UPDATE BRANCH, PIN IC8.2
```

will cause the process to halt only if two or more active paths are encountered (a BRANCH) or if IC8 pin 2 is encountered.

CONNECT

This command may be used to output a final listing of the paths taken to generate equations. It is similar to the final display for TRACK or BACK and can be used with any Initiate Process Start Command. The command may be issued *after* completing an Initiate Process Start command (such as EQUATION). It is also possible to terminate a run in progress (through use of CTRL X) and then issue the CONNECT command *next*, if it is desired to examine the paths taken prior to generating a complete EQUATION run.

EXIT

This command *closes* any open files that are being created for output listings (if O=DSKn:NAME), and then *exits* to the APG Monitor top level. The command differs from the Teletype CTRL C command in that CTRL C may be used to exit immediately, *without* closing any open output files.

B.3.3.4 Command Examples

Commands may be issued separately, (with the Initiate Process Start Commands (TRACK, BACK, and EQUATION) being issued last, after issuing all other required commands. It is also possible to combine a group of commands on the same line if each command is separated by a semi-colon (;) and the Initiate Process Start Command is the last command in the sequence. All commands may be shortened to include only the underlined characters listed in Table B-2. In addition, entire clauses may be dropped from the Initiate Process Start Commands if they are not required. For example, the only required segments might be:

```
TRACK p
BACK  p
EQUATION  p
```

It should be noted, however, that the last arguments entered will be used. If no prior arguments have been entered (or the command mode has just been entered from the APG Monitor top level) then the default values listed in Figure B-4 will be assigned.

The following examples of E/T/C Mode Command lines are provided for a users convenience:

Command	Function
F FI CO; U N;S 180, T IC7.5 ↓	Track back from IC7.5 at PRETTY test Step 180. Do not provide any updated displays. Output a final display that lists both active paths taken and termination points.
T IC3.4 ↓	If this command is issued next, the software will track back from IC3.4. Since the FORMAT and STEP commands have not been changed, they will remain as they were, and a final display that lists both active paths taken and termination points will be output.

Table B-2

COMMAND ABBREVIATIONS

The following set of underlined characters reflect the minimum word length of each command word:

ACTIVE
ALL
AT
BACK
BRANCH
COMPLETE
CONNECT
ENDS
EQUATION
EXIT
FINAL
FORMAT
FORWARD
FULL
GENERATE
LEVEL
LIST

MEMORY
MSI
NONE
ONE
PATH
PIN
SIMPLE
STATUS
STEP
SUMMARY
TABLE
TRACK
UNTIL
UPDATE
ZERO

`F FI SU; T IC2.1 ↓`	If this command is issued next, the software will track back from IC2.1 at the test STEP listed from the previous command (180). A final display that includes only the final termination points (SUMMARY) will be output.
`U LE; F UP AL; S 0; B IC8.2 UN LE 6 ↓`	Run BACK connectivity from IC8.2 until the sixth LEvel of logic is reached. UPdate at each LEvel, and display all Inputs at that point (whether active or not).
`U N; F EQ TA; E IC2.1 ONE ↓`	Run Equation Generation. List all terms required to set IC 2.1 High (ONE). Terms are listed in a tabular format (TABLE), and only the final table will be output since <u>UP</u>DATE <u>AT</u> <u>N</u>ONE has been specified.

B.3.4 <u>Output Formats</u>

Updated and final display outputs for both TRACK and BACK follow the same general format. Final display outputs normally provide more information than updated outputs, however, the format is identical. EQUATION, on the other hand, is presented in an entirely different format from either TRACK or BACK as EQUATION allows the selection of either a listing of terms or a table of terms. Output formats for both major categories are described in the following paragraphs.

B.3.4.1 <u>Tracking and Back Connectivity Output Formats</u>

A typical example of a final tracking display appears in Figure B-6 The major difference between an updated (interim) display and a final display is that the total amount of data provided in the updated (interim) display will normally be less than in the final display. The format used will be identical. Each display may include the following types of information:

1. A Header (STATUS) message that identifies the type of display being output and lists the UUT name as the following two examples indicate:

```
STATUS:  ETC LEVEL UPDATE ON M8091

STATUS:  ETC FINAL ON M8091
```

2. The IC Input or Output pin, EXTernal Name, or Intermediate Name at which the action <u>started</u> (track back <u>from</u> this point). This information appears at the top of the form in the starred window.

```
  * * * * * * * * * * * * *
* E17.3                         *
  * * * * * * * * * * * * *
```

The starred window may also include the Good Board Simulation Results (PRETTY) File test step at which activity <u>first</u> occurred together with the type of activity.

```
  * * * * * * * * * * * * *
* E17.3 01 AT 250             *
  * * * * * * * * * * * * *
```

In the example shown, E17.3 went from a Low (0) to a High (1) state at test step 250. Test step 250 was the <u>first</u> test step <u>prior</u> to

```
STATUS:  ETC FINAL ON M8091  AT LEVEL 5

             *********************************************************
            *                                                         *
           *            E17.3         01  AT  250                      *
            *                                                         *
             *********************************************************

                 E17.3*      E17.3      01  AT  250                   SOURCE NODE

                 E22.3*      E17.1      10  AT  250
/MUL/            E22.3*      E17.2      10  AT  250                     E17.3*

/NTR/            E47.6*      E22.1      01  AT  161
                 E27.4*      E22.4      01  AT  250
                 C76.2*      E22.9      01  AT  250
                 E22.2*      E22.2      01  AT  250                     E22.3*

                 E59.6*      E27.1      10  AT  250
/NTR/            GND.1*      E27.15     X0  AT  0
/NTR/            E41.3*      E27.2      01  AT  248
/NTR/            E26.10*     E27.3      10  AT  248                     E27.4*

               WIRE-AND*     C76.1      01  AT  250                     C76.2*

/NTR/            E47.6*      E22.1      01  AT  161
/MUL/            E27.4*      E22.4      01  AT  250
/MUL             C76.2*      E22.9      01  AT  250
/MUL/            E22.3*      E22.3      10  AT  250                     E22.2*

                 E48.8*      E59.3      01  AT  250
/NTR/            E63.11*     E59.4      01  AT  197
/NTR/            E63.6*      E59.5      01  AT  248                     E59.6*

/NTR/            R76.1*    WIRE-AND     X1  AT  0
                 E75.13*   WIRE-AND     01  AT  250                   WIRE-AND*

/EXT/            DE2 #122*   E48.10     01  AT  250
/NTR/            E64.6*      E48.9      01  AT  248                     E48.8*

/MUL/            E59.6*      E75.11     10  AT  250
/NTR/            E55.8*      E75.12     01  AT  226                     E75.13*
```

Figure B-6. A Typical Final Tracking Display.

the one specified by the STEP Command to show <u>activity</u>. The Tracking software tracked back from PRETTY Test Step 250.

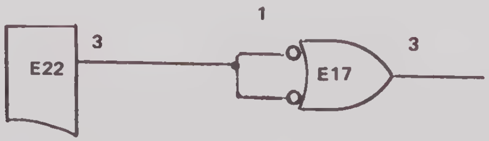
3. The first listing of the display normally identifies the Source Node or the node at which tracking (or connectivity) is to begin. In the example

```
        E17.3*   E17.3  01  AT  250  SOURCE NODE
```

the source node (E17.3) is also a driving pin (*).

4. Other listings can identify circuit paths and/or final points being tracked (depending on the type of FORMAT commands issued and whether updated or final displays are being output). In the following example,

```
        E22.3*   E17.1  10  AT  250
/MUL/   E22.3*   E17.2  10  AT  250  E17.3*
```



the active paths taken are listed. Note that E22.3 is the pin driving E17, that MULtiple paths are involved, and that both paths to E17.1 and E17.2 went from a High (1) to a Low (0) state at PRETTY test step 250. In this example, the input (E17.3) being tracked back from is the source node. The abbreviation MUL identifies the type of termination condition encountered. A complete set of identifiers that may appear in TRACK or BACK displays are listed in Table B-3. Note again, that the contents of each display may differ depending on the type of display commands issued.

Table B-3

TRACKING AND BACK CONNECTIVITY DISPLAY IDENTIFIERS

Identifier	Meaning
	TRANSITIONAL IDENTIFIERS
X	Unknown State.
P	A Positive Pulse ending Low has occurred.
N	A Negative Pulse ending High has occurred.
0	Low State.
1	High State.
X0	Lists pin activity. In this example, a transition from an Unknown (X) to a Low (0) State has occurred.
	TERMINATION IDENTIFIERS
/EXT/	An EXTernal pin has been encountered and tracking cannot proceed along this path.
/FBK/	A Feedback Loop (FBK) has been identified that is not to be tracked.
/LEV/	The LEVel specified by the UNTIL LEVEL n command has been reached and tracking cannot proceed beyond this point.
/MUL/	Identical (MULtiple) paths have been identified and will not be tracked further.
/NEQ/	The circuit point cannot be expanded further.
/NTR/	This node is not tracked as it does not match the activity being tracked.
/UNC/	An UNCconnected path has been encountered that cannot be tracked further.

B.3.4.2 Equation Generation Output Formats

Equation generation listings may be output as either a list of terms or a table or terms (depending on the FORMAT Command selected). Both formats are shown in Figure B-7. Each format provides the following information:

1. A header (STATUS) message that identifies the type of display being output and lists the UUT name.

2. The IC Input or Output pin, EXTernal Name, or Intermediate Name of the circuit point or node that is to be set High or Low by the set of equations being generated. This information appears at the top of the form in the starred window.

```
STATUS:  EQUATION GENERATION ON M8091  AT LEVEL 5

          *******************************************************
         *                                                       *
        *            E69.10       X0  AT  0                       *
         *                                                       *
          *******************************************************

SET TO ZERO
                      01234 56789  0123
   DF2 #123           HHHHH H::HH  HHHH
   DH2 #124           :HH:H HHHHH  :HH:
   EA1 #136           HHHHH H::::  H:::
   EB1 #137           HHHHH H::::  H:::
   EC1 #138           HHHHH H::::  H:::
   ED2 #154           HHHHH H::::  H:::
   EE1 #140           HHHHH H::::  H:::
   EE2 #155           HHHHH H::::  H:::
   EU1 #151           :LL:L LLLLL  :LL:
   EV1 #152           :HH:H HHHHH  :HH:
   E51.8              :L::: :L:L:  :L::
   E52.8              :LLLL LLLLL  LLLL

                                   TABLE FORMAT
```

```
          *******************************************************
         *                                                       *
        *            E69.10       X0  AT  0                       *
         *                                                       *
          *******************************************************

SET TO ZERO

   TERM 0
        DF2 #123      WIRE-AND      WIRE-AND       EE2 #155      ED2 #154
        EA1 #136        EB1 #137         EC1 #138      EE1 #140

   TERM 1
        /E51.8        DH2 #124      /EU1 #151      DF2 #123      EV1 #152
      WIRE-AND          EE2 #155         ED2 #154      EA1 #136      EB1 #137
          EC1 #138        EE1 #140

   TERM 2
        /E52.8        DH2 #124      /EU1 #151      DF2 #123      EV1 #152
      WIRE-AND          EE2 #155         ED2 #154      EA1 #136      EB1 #137
          EC1 #138        EE1 #140

                                   LIST FORMAT
```

Figure B-7. Equation Generation Output Formats.

```
   * * * * * * * * * * * * *
 *    E69.10    X0   AT   0    *
   * * * * * * * * * * * * *
```

The starred window may also include the Good Board Simulation results (PRETTY) file test step at which activity occurred or the first test step prior to the one listed that has activity. This information applies to Tracking only, but may also appear in Equation Generation Outputs if a Good Board Simulation run has been performed.

3. This message provides an indication of what purpose the equation is intended for. For example, the High or Low state that the pin or node listed (E69.10) is to be driven to by the terms being presented.

```
          SET TO ZERO
```

In the example shown, E69.10 is to be driven Low (ZERO) by the terms listed.

4. Terms are sets of equations that will drive the pin or node listed into the High or Low state specified. Terms are presented in sequence for lists and in columns for tables.

5. Terms for lists always indicate the High state. For example, assume that E21.3 is to be driven Low. The following set of terms

```
TERM 0
     /E17.3   E18/4   E20/3

TERM 1
     E19.1   /E16.5   E11.2
```

would indicate that either (E17.3 and E18.4 and E20.3) or (E19.1 and E16.5 and E11.2) must be set High in order to set E21.3 Low. Note that the / symbol is used to specify the barred (NOT) condition (if E17.3 is set High, then E17.3 must be set Low in order to satisfy the terms of the equation.)

6. Terms for Tables do not list the NOT condition. For example, the table generated terms required to set E21.3 to Zero might be:

```
               0 1 <----COLUMNS OF TERMS

          E17.3  L :
          E18.4  H :
          E20.3  H :
          E19.1  : H
          E16.5  : L
          E11.2  : H
```

Where a colon (:) indicates that the state of the pin is unchanged.

7. Equations cannot be generated for any IC elements with memory.

Log Mode-Appendix C

C.1 INTRODUCTION

The LOG Mode provides the facility for logging data learned during DIAGNOSE Mode. This data is stored in pre-allocated files and may be read out later in machine readable language.

The logging facility consists of four logical parts:

Log File Initializer
Log Data Collector
Log File Report Generator
Log File Status Display

Before the LOGging mode is entered, a working test program (.TP) and a stable network description (.SR) must be developed. When this is accomplished the data logging file may be created by entering the Log File Initializer Sub-Mode. A separate log file must be initialized for each CAPS section to be logged. At this time the user may specify which items are or are not to be logged. When DIAGNOSIS Mode is run, CAPS automatically stores the specified information in the appropriate LOGging (.DL) file.

When it is desired to read the stored information, the Print Sub-Mode is enabled. The information is printed in ASCII code on the specified output device.

The Log File Status Display Sub-Mode allows the user to check the status of a log file. By accessing this sub-mode the user can see a comprehensive summary of the specified file.

C.2 PREREQUISITES

The LOGging Mode is a mode that has certain prerequisites that must be fulfilled before it can be run. In addition to a working .TP and a stable .SR, certain options in the DIAGNOSIS Mode must be enabled.

They are:

L=	which enables or disables the logging function if a logging file has been initialized.
C=	which enables or disables operator comments
LD=	which specifies the location (device) of the log file.

Terms

The word "ITEM" as it is used in the description of LOGging Mode pertains to those words listed in Table C-1. Table C-1 is a description of each item that may be saved in the .DL file and which selection variables (see Report Generator Section) require them to be present. Other terms used in this description are defined as follows:

TABLE C-1

ITEM DEFINITION

ITEM	DESCRIPTION
DATE	The date the diagnosis was made: the date last entered with the OS/8 DATE command. Selection variable: DATE.
DIAGNOSIS	The diagnosis source. For guided probe diagnoses, this also enables saving extra information, including a more specific diagnosis and suspected failure points. Selection variables: DIAGNOSIS.
FFT	(First Failing Test). The CAPS test number at which the test program originally failed. For analog failures, the last CAPS test number preceding the failing analog test step. Selection variable: FFT.
FFTV	(First Failing Test Vector). A vector containing the numbers of driver/sensors which were in error at the first failing test. Selection variable: FFTV
FBTV	(Fail-by-Test Vector). A vector describing those test steps after the FFT which also failed. Selection variable: FBTV.
COMMENT	(Operator Comment). A comment (such as serial number) which may be entered by the operator at the time the diagnosis is made. Selection variable: COMMENT.
HISTORY	(Probe History). The probe history file in a compact form. (Guided probe diagnoses only). Selection variable: HISTORY.
CLASS	(Fault Classes). Fault classes detected by Automatic Fault Location (AFL). (AFL initial and final diagnoses only). Selection variable: CLASS.

RECORD	the collection of all items saved for one diagnosis
DIAGNOSIS SOURCE	The diagnostic mode making the diagnosis for a particular record
FORTRAN	for a more complete explanation of the language refer to the O/8 HANDBOOK published by Digital Equipment Corporation.

C.3 ACCESSING LOGGING MODE

The LOGging Mode is accessed from the CAPS Monitor page using any of the procedures specified in Section 3 of the CAPS Manual. For example:

LOG and XMIT

When accessed, the first of five LOG option pages is displayed. To view succeeding pages depress the XMIT pushbutton.

C.4 OPTION PAGES

FILE USAGE and SUB-MODE SELECTION Page (see Figure C-1)

There are three entries that must be made on this page. The first is L=(Log File). Here the user may specify the device the file is stored on, the file name, which is the same as the CAPS section for which PREPARE Mode has been run, and the .DL extension.

For example:

```
L=DSK:GRDEM.DL
```

For the L= option it is necessary to specify just the file name. The system automatically assigns DSK: as the device and the .DL extension.

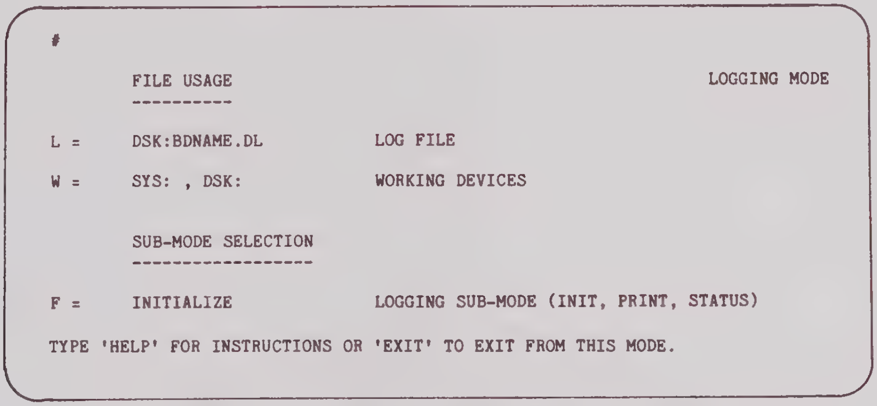
The second entry is W=(Working Devices). This entry specifies the working devices previously used when running the PREPARE Mode for the CAPS section corresponding to L=. For example:

```
W=DSK0:,DSK1:
```

The last entry on this page is F=(Logging Sub-Mode). This entry specifies the sub-mode to be run during the LOGging Mode. There are three alternatives: INITialize, PRINT, or STATUS.

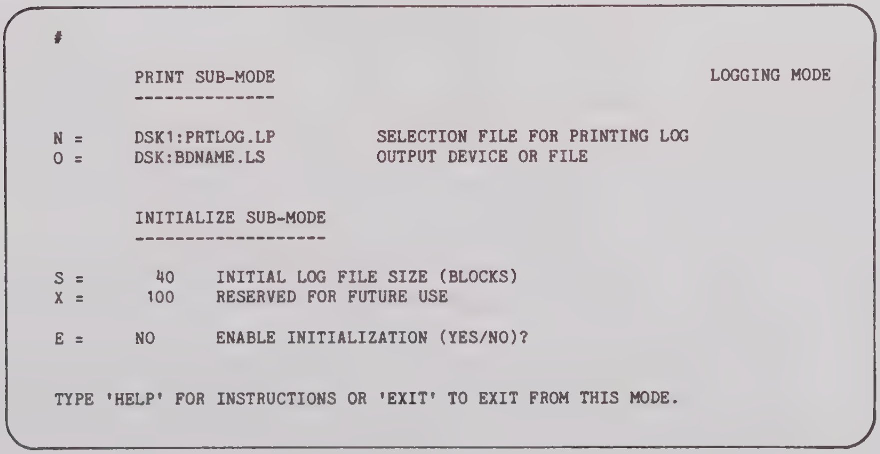
PRINT and INITIALIZATION Options Page (see Figure C-2)

There are five entries that can be made on this page. The first two entries apply to the PRINT Sub-Mode and the last three entries apply to the INITIALIZE Sub-Mode.



```
#

         FILE USAGE                                          LOGGING MODE
         ----------

L =      DSK:BDNAME.DL              LOG FILE

W =      SYS:  , DSK:               WORKING DEVICES

         SUB-MODE SELECTION
         ------------------

F =      INITIALIZE                 LOGGING SUB-MODE (INIT, PRINT, STATUS)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure C-1. FILE USAGE and SUB-MODE SELECTION Page.



```
#

         PRINT SUB-MODE                                      LOGGING MODE
         --------------

N =      DSK1:PRTLOG.LP             SELECTION FILE FOR PRINTING LOG
O =      DSK:BDNAME.LS              OUTPUT DEVICE OR FILE

         INITIALIZE SUB-MODE
         -------------------

S =        40     INITIAL LOG FILE SIZE (BLOCKS)
X =       100     RESERVED FOR FUTURE USE

E =      NO       ENABLE INITIALIZATION (YES/NO)?

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure C-2. PRINT and INITIALIZE Page.

C.4.1 PRINT Sub-Mode Options

N=(Selection File for Printing Log)

With this entry the user specifies the input report generator program. This may be specified as device, file name, and extension. For example:

DSK:GRDEM.LP

The user may specify only the file name and the system automatically supplies the location and extension.

O=(Output Device or File)

With this entry the user specifies the device or file name where the output of PRINT Sub-Mode is stored.

C.4.2. INITIALIZE Sub-Mode Options

The following three entries apply to the INITIALIZE Sub-Mode.

S=(Initialize Log File Size)

Here the user specifies the size of the LOG file blocks. There is no predetermined way to establish the number. It is suggested that the number 400 be used initially. There is no guarantee that 400 will be sufficient, so the user should periodically check the Status of the LOG file to verify the amount of information it contains. As the user becomes more famaliar with the operation of this mode and better understands what information he plans to store in the LOG file he will be better able to specify this number.

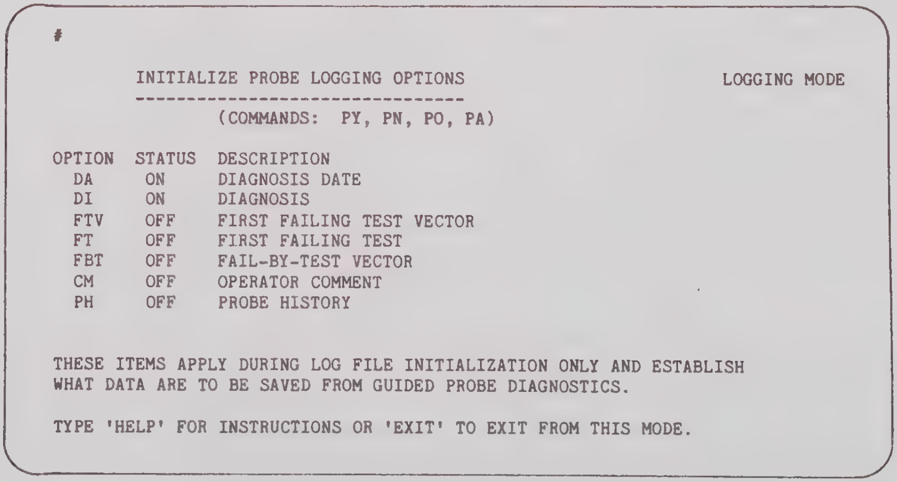
X=(Reserved for future use)

This is self-explanatory.

E=(Enable INITIALIZATION YES/NO)

The entry enables (YES) or disables (NO) initialization. This option is used in conjunction with the F= option on the first option page. It acts as an interlock to prevent the undesired initialization of a log file. It is automatically reset to NO after the INITIALIZE Sub-Mode is run.

INITIALIZE PROBE LOGGING OPTIONS Page (see Figure C-3)



```
#

        INITIALIZE PROBE LOGGING OPTIONS                         LOGGING MODE
        --------------------------------
                (COMMANDS:  PY, PN, PO, PA)

OPTION  STATUS  DESCRIPTION
  DA     ON     DIAGNOSIS DATE
  DI     ON     DIAGNOSIS
  FTV    OFF    FIRST FAILING TEST VECTOR
  FT     OFF    FIRST FAILING TEST
  FBT    OFF    FAIL-BY-TEST VECTOR
  CM     OFF    OPERATOR COMMENT
  PH     OFF    PROBE HISTORY


THESE ITEMS APPLY DURING LOG FILE INITIALIZATION ONLY AND ESTABLISH
WHAT DATA ARE TO BE SAVED FROM GUIDED PROBE DIAGNOSTICS.

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure C-3. INITIALIZE PROBE LOGGING OPTIONS Page.

This page applies only during log file initialization and establishes the data to be saved from guided probe diagnostic operations.

DA (Diagnosis DATE)

This option allows the date of the diagnosis to be saved.

DI (Diagnosis)

This option allows the diagnosis to be saved.

FTV (First Failing Test Vector)

This option allows the logging program to save a vector containing the numbers of the driver/sensors which were in error at the first failing test.

FT (First Failing test)

This option allows the logging of the CAPS test number at which the test program originally failed. For analog failures, the last CAPS test number preceding the failing analog test step is logged.

FBT (Fail-By-TEST Vector)

This option allows the logging of a vector containing failing test step numbers after the first failing test step.

CM (Operator Comment)

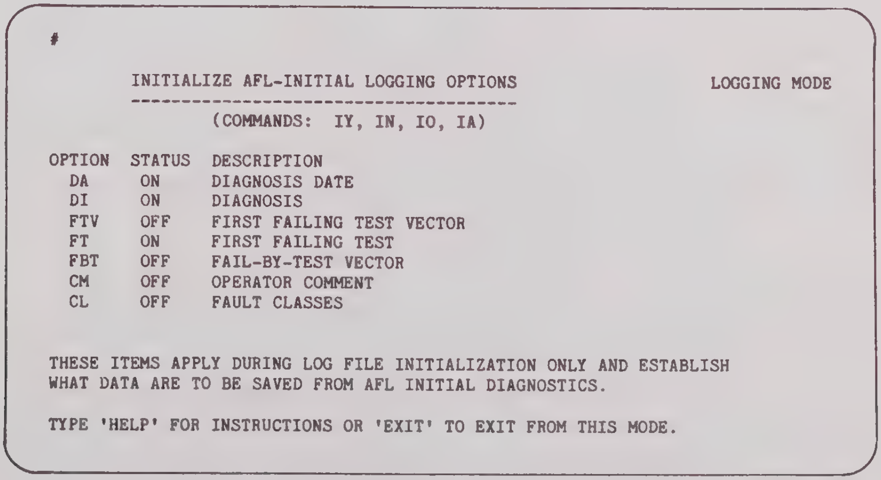
This option allows the logging of a comment (such as a serial number) that may be entered by the operator at the time the diagnosis is made.

PH (Probe History)

This option allows the logging of the probe history file in a compact form.

INITIALIZE AFL-INITIAL LOGGING OPTIONS Page (see Figure C-4)

This page applies during log file initialization only and establishes the data to be saved from AFL-initial diagnostics. These options are the same as the options described in the INITIALIZE PROBE LOGGING OPTIONS Sub-Section except for CL.



```
#

        INITIALIZE AFL-INITIAL LOGGING OPTIONS                  LOGGING MODE
        --------------------------------------
                (COMMANDS:  IY, IN, IO, IA)

OPTION  STATUS  DESCRIPTION
  DA     ON     DIAGNOSIS DATE
  DI     ON     DIAGNOSIS
  FTV    OFF    FIRST FAILING TEST VECTOR
  FT     ON     FIRST FAILING TEST
  FBT    OFF    FAIL-BY-TEST VECTOR
  CM     OFF    OPERATOR COMMENT
  CL     OFF    FAULT CLASSES


THESE ITEMS APPLY DURING LOG FILE INITIALIZATION ONLY AND ESTABLISH
WHAT DATA ARE TO BE SAVED FROM AFL INITIAL DIAGNOSTICS.

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure C-4. INITIALIZE AFL-INITIAL LOGGING OPTIONS Page.

CL (Fault Classes)

This option allows logging of fault classes that are detected during an AFL initial diagnosis.

INITIALIZE AFL-FINAL LOGGING OPTIONS Page (see Figure C-5)

This page applies during log file initialization only and establishes the data to be saved from AFL-final diagnostics. The options are the same as the options described in the INITIALIZE PROBE LOGGING OPTIONS Sub-Section except for CL which is explained in the INITIALIZE AFL-INITIAL LOGGING OPTIONS Sub-Section.

C.5 FILE INITIALIZATION

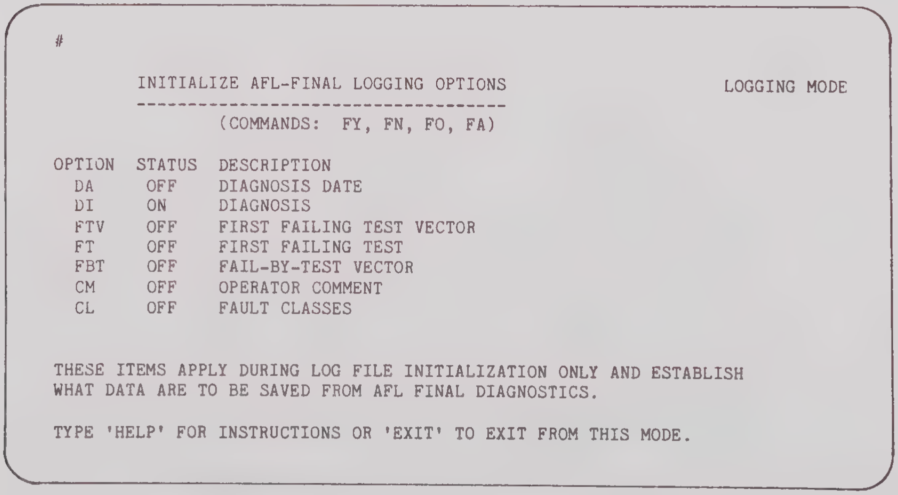
After the LOGging Mode is accessed and the user has properly filled in the Monitor pages he may initialize the log file. A data logging file (.DL) must be initialized for each CAPS section for which information is to be logged. Each .DL file is named with the exact name used in the CAPS section.

The initial log file size must be adequate to contain all the logged information. This number can only be determined with experience and is dependent on both the type and amount of information to be stored. If, however, the file should become full an error message will result and further logging for that CAPS section will be disabled. The error message is:

```
LOG FILE device:name.DL IS FULL. FURTHER LOGGING DISABLED
```

The user should periodically check the Status of the Log file to ensure that it is not going to become full. There are some items that require more space than others. The size of these items is variable but the items that are significant are:

FFTV
FFBTV
HISTORY
CLASS
DIAGNOSIS (for probe diagnosis)



```
#

        INITIALIZE AFL-FINAL LOGGING OPTIONS                    LOGGING MODE
        ------------------------------------
                (COMMANDS:  FY, FN, FO, FA)

OPTION  STATUS  DESCRIPTION
  DA     OFF    DIAGNOSIS DATE
  DI     ON     DIAGNOSIS
  FTV    OFF    FIRST FAILING TEST VECTOR
  FT     OFF    FIRST FAILING TEST
  FBT    OFF    FAIL-BY-TEST VECTOR
  CM     OFF    OPERATOR COMMENT
  CL     OFF    FAULT CLASSES


THESE ITEMS APPLY DURING LOG FILE INITIALIZATION ONLY AND ESTABLISH
WHAT DATA ARE TO BE SAVED FROM AFL FINAL DIAGNOSTICS.

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE.
```

Figure C-5. INITIALIZE AFL-FINAL LOGGING Page.

NOTE

The Initialize Sub-Mode of LOGging is equipped with a type of interlock. This interlock prevents the user from inadvertantly initializing a file. Both the Sub-mode selection option F= must be specified as INIT and the Enable initialization option must be enabled (E=YES) before initialization can occur.

After the file is initialized the E= option is automatically turned off (E=NO).

C.6 LOG DATA COLLECTION

The collection of data to be stored in the log file is automatically accomplished during DIAGNOSIS Mode.

NOTE

The following three options are found on the LOGGING page of DIAGNOSIS Mode.

Before DIAGNOSIS Mode is run the logging enable option must be enabled (L=YES), the user must decide if comments by the operator are allowed (C=), and the device on which the log file is located must be specified (LD=).

When the DIAGNOSIS Mode is run, failures are automatically logged in the appropriate .DL file when all the following conditions are met:

1. Logging was enabled in DIAGNOSIS Mode
2. Logging was initialized --- a .DL file with the proper name exists and at least one item was enabled in that .DL file for the diagnosis source which detected the error.
3. The .DL file was not previously marked FULL and will not be over-filled with the new record.

If the user has enabled operator comments both in the appropriate LOG File Monitor Page (CM=) and the LOGGING Page of DIAGNOSIS Mode, the one line comment may be entered on either TEC (CRT) or the Teletype. The prompt to enter the comment appears on the second line of the TEC (CRT):

```
***ENTER DIAGNOSIS LOG COMMENT ABOVE OR ON TELETYPE
```

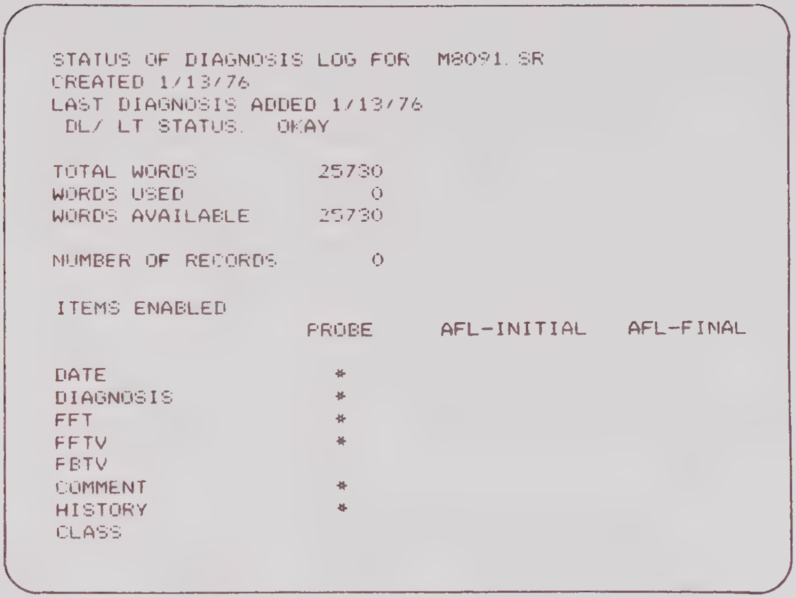
It appears after the Diagnosis information is presented on the visual terminal and the teletype bell is rung. XMIT on the visual terminal or CR on the teletype ends the comment. The comment has a maximum length of one line.

NOTE

The symbol @ is not permitted in comments.

C.7 LOG FILE STATUS DISPLAY

The user at any time may check the status of a Log file by setting the function option to status (F=S) and RUNning the LOG Mode. The status information is displayed on the visual terminal (see Figure C-6). Most of the status information is



```
STATUS OF DIAGNOSIS LOG FOR  M8091.SR
CREATED 1/13/76
LAST DIAGNOSIS ADDED 1/13/76
 DL/ LT STATUS.   OKAY

TOTAL WORDS            25730
WORDS USED                 0
WORDS AVAILABLE        25730

NUMBER OF RECORDS          0

 ITEMS ENABLED
                       PROBE     AFL-INITIAL   AFL-FINAL

 DATE                    *
 DIAGNOSIS               *
 FFT                     *
 FFTV                    *
 FBTV
 COMMENT                 *
 HISTORY                 *
 CLASS
```

Figure C-6. Log File Status Display.

self-explanatory, but the entry ".DL/.LT STATUS" needs further explanation. The .DL/.LT status may be OKAY, UNKNOWN, or CONFLICT. OKAY indicates that the LOG File is consistent with other program files. UNKNOWN indicates that the .LT file was not available for checking. CONFLICT indicates that the user's program files (.SR, .TP) have changed in a way that <u>may</u> have made them inconsistent with the LOG File. Therefore, the logging software refuses to add further diagnosis information to the LOG File (see cautions).

C.8 LOG FILE REPORT GENERATOR (see Figure C-7)

The Log File Report Generator is a user produced program that allows the user selectivity output information in machine-readable form.

To output the information selected and formatted by the Log File Report Generator the user must run the PRINT Sub-Mode of LOGging (F=PRINT). When specifying PRINT the user must also specify the Output File (O=) and the Report Generator Program (N=).

Employing the EDIT Mode the user must first produce the File Report Generator program. The simplicity of this program results from the fact that there are only four basic statements used in its development. The four statements are:

PRINT (output)
IF (conditional)
BEGIN .... FINISH (block Structure)
QUIT (control)

<u>PRINT Statement</u>

The PRINT Statement consists of the keyword PRINT followed by one or more Print Elements, separated by commas (,). A Print Element may be any item or a literal string. For example:

```
PRINT FFT, FFTV, 'END OF RECORD';
```

Any character except end-of-file may appear between the apostrophes (') of a literal string, including (especially) carriage return. To include an apostrophe in the string itself, two are used, as in

CAN''T

which represents

CAN'T.

Literal strings are printed as they appear. Each item is output according to its own fixed format.

```
   THIS REPORT GENERATOR PROGRAM PRINTS THE CONTENTS OF
   THE ENTIRE LOG FILE
   IF AN ITEM IS PRESENT, ITS NAME IS PRINTED,
   PRECEDED BY A ´+´, FOLLOWED BY A
   WITH ITS VALUE    STARTING ON THE NEXT LINE
   HOWEVER, IF THE ITEM IS ABSENT, ITS NAME IS
   PRINTED, PRECEDED BY A  -

IF DATE PRESENT
     THEN PRINT ´+DATE:
   DATE
     ELSE PRINT ´-DATE

IF DIAGNOSIS PRESENT
     THEN PRINT ´+DIAGNOSIS:
   DIAGNOSIS
     ELSE PRINT ´-DIAGNOSIS

IF FFT PRESENT
     THEN PRINT ´+FFT:
   FFT
     ELSE PRINT ´-FFT

IF FFTV PRESENT
     THEN PRINT ´+FFTV.
   FFTV
     ELSE PRINT ´-FFTV

IF FBTV PRESENT
     THEN PRINT ´+FBTV:
   FBTV
     ELSE PRINT ´-FBTV

IF COMMENT PRESENT
     THEN PRINT ´+COMMENT
   COMMENT
     ELSE PRINT ´-COMMENT

IF HISTORY PRESENT
     THEN PRINT ´+HISTORY:
   HISTORY
     ELSE PRINT ´-HISTORY

IF CLASS PRESENT
     THEN PRINT ´+CLASS.
   CLASS
     ELSE PRINT ´-CLASS

PRINT ´*** END OF DIAGNOSIS ***

   EACH SET OF ITEMS FOR ONE DIAGNOSIS IS FOLLOWED
   BY THIS SEPARATOR
```

Figure C-7. Typical Log File Generator Program.

IF Statement

The IF Statement is the mechanism which allows selective printing of log information. Its form is:

```
                IF Boolean expression THEN statement;
      or
                IF Boolean expression THEN statement ELSE statement;
```

"Boolean expression" is an expression which has a value of TRUE or FALSE. It is expressed in the usual way and is fully explained in the Boolean Expression Subsection. When the Boolean expression evaluates to TRUE, the statement following THEN (the "THEN clause") is executed. When the Boolean expression evaluates to FALSE, the ELSE clause is executed, if there is one.

A "statement" may be any of the four statement types. Note particularly that the BEGIN...FINISH construction allows this "statement" to perform arbitrarily complex computations. The only restriction here is that any IF statements in the "THEN" clause may not be IF-THEN-ELSE statements, only IF-THEN. Thus, a statement of the form:

```
IF ... THEN ... IF ... THEN ... ELSE ...
```

would be interpreted as having its "ELSE" clause associated with the <u>first</u> IF... THEN.

BEGIN...FINISH Statements

These two statements bracket a cluster of one or more other statements into a "block". A block is, itself, considered one statement, and blocks may be nested to arbitrary depths. For example:

```
BEGIN

   PRINT FFT;

   PRINT COMMENT;

FINISH;
```

is equivalent to

```
PRINT FFT, COMMENT;

or

BEGIN

   PRINT FFT, COMMENT;

FINISH;
```

QUIT Statement

The QUIT statement terminates execution of the current block. No further statements in the block are executed. For example, in the (useless) program fragment:

```
BEGIN

   PRINT FFT;

   QUIT;

   PRINT COMMENT;

FINISH;
```

the QUIT would prevent the second PRINT from ever being performed. More commonly, QUIT is used in an IF statement, as in:

```
BEGIN

   IF DATE < 3/9/79  THEN QUIT;

   PRINT DATE, FFT;

FINISH
```

Boolean Expressions

Boolean expressions derive their values from a number of relational operators (relations). Most of the relations compare two numbers, two sets, or two strings and produce a Boolean (TRUE/FALSE) value. In each case one operand is always a "selection variable", and the other is always a parameter or parameters specified in the report generator program. "Selection variables" are simply pre-named variables which stand for the corresponding item in the log record currently being processed. For example, the selection variable FFT has the value 117 if the first failing test number saved in the current log record is 117.

It is possible to refer to selection variables in the report generator program which, at log initialization time, were not enabled for saving. In such cases, a relation referring to a non-saved selection variable is _always_ FALSE. The presence or absence of an item may also be tested.

Arithmetic Relation Operators

Four selection variables use arithmetic relational operators. The operators are:

=	equal to
#	not equal
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

Numeric Selection Variables

The four numeric selection variables are DATE, FFT, DIAGNOSIS, HISTORY.

DATE Relation Date

A 'date' is expressed as month/day/year, as in 3/9/79. The expression DATE> 9/11/70 is TRUE if the data saved in the current log record is later than March 11, 1970, and FALSE if the logged date is equal to or before that date.

FFT Relation Step

A 'step' is either expressed as an integer, meaning a CAPS test number, or an integer followed by a colon (:), meaning the CAPS test number corresponding to the specified statement label in the test program. The obvious comparison is performed.

DIAGNOSIS Relation Integer

Each type of diagnosis is assigned a number (see Table C-2). The recorded diagnosis type is compared with the integer.

HISTORY Relation Integer

The probe history consists of successive probed points. This selection variable has the value of the "length" of the probe history in terms of points. The length is compared against the integer. This variable is useful for locating long probing sequences which might be aided by starting or jumping points.

Some examples of arithmetic relational expressions are:

DATE <= 2/8/79	dates before February 8, 1979
FFT > 153:	first failing tests past statement label 153
DIAGNOSIS # 300	diagnoses not from AFL-final
HISTORY < 8	probe histories shorter than 8 points

Selection Variables

Five selection variables use set relational operators. The general form of the set relations is:

selection variable set relation {argument set}.

The set relations are:

SUBSET	
SUBSET OF	TRUE if all members of the selection variable set are also members of the argument set. FALSE otherwise.
SUPERSET	
SUPERSET OF	TRUE if all mambers of the argument set are also members of the selection variable set. FALSE otherwise.
INTERSECTS	TRUE if at least one member of the selection variable set is also a member of the argument set. FALSE otherwise.
EQUALS	TRUE if and only if each member of the selection variable set is also a member of the argument set, and *vice versa*. FALSE otherwise.

Table C-2

DIAGNOSIS NUMBER

NUMBER	DEFINITION
100	Guided probe (Diagnosis not saved)
101	PROBE TERMINAL POINT
102	BROKEN CONNECTION
103	BAD FEEDBACK LOOP
104	DRIVER ERROR
105	NODE FAILURE
106	WIRED-AND NODE FAILURE
107	ANALOG - NO DIGITAL CAUSE
108	ANALOG FAILURE - CAN'T CONTINUE
109	TWO NODES SHORTED
110	SUSPECTED TRACK SHORT
111	PIN FAILURE/BAD IC
112	BAD CONN. - PROBABLE MISPROBE
113	SHORT TO VCC
114	SHORT TO GND PLANE
117	Same as 101 but with no DRM improvement
119	Same as 103 but with no DRM improvement
120	Same as 104 but with no DRM improvement
121	Same as 105 but with no DRM improvement
122	Same as 106 but with no DRM improvement
200	AFL initial
300	AFL final

The argument set varies with each selection variable. The five selection variable sets are FFTV, FBTV, CLASS, DEVICE, PIN.

FFTV Set Relation {Argument Set}

This selection variable corresponds to the set of failing driver/sensor numbers for the first failing test. The argument set, therefore, expresses driver/sensor numbers. The driver/sensors may be referred to symbolically by their external names ('EXT1'), by number ('.4'), or as a range of numbers ('.4 TO .8'). Thus

FFTV SUBSET OF {.9 TO .11, EXT1, EXT2, .20}

would be true if the set of failing driver/sensors at the first failing test is a subset of driver/sensors 9, 10, 11, 20, and the two whose names are EXT1 and EXT2.

FBTV Set Relation {Argument Set}

This selection variable corresponds to the set of CAPS test step numbers which failed after the first failing test. Hence the argument set expresses CAPS test step numbers. The step numbers may appear as statement labels ('251:'), which correspond to CAPS steps, CAPS step numbers ('33'), or ranges of CAPS step numbers ('33 TO 85'). The special CAPS step number LAST stands for the number of the last digital test. For example,

FBTV SUPERSET {10, 13, 85 TO LAST, 115:}

would be true if the set of failing step numbers included steps 10, 13, 85 to the last digital test, and the step whose label is 115.

CLASS Set Relation {Argument Set}

This selection variable corresponds to the set of fault classes selected by the automatic fault location (AFL) diagnostic mode. The argument set describes fault classes and ranges of fault classes. For example,

CLASS INTERSECTS {5, 280, 312 TO 314}

is TRUE if any fault class detected by AFL is one of 5, 280, 312, 313, or 314.

DEVICE Set Relation {Argument Set}

This selection variable corresponds to the set of devices (IC's) appearing in a guided probe diagnosis as possible failures. The IC's are referred to by name as in

DEVICE SUBSET {IC10, IC15}

or

DEVICE EQUALS {IC3}.

The second example would be TRUE if the only IC appearing in the diagnosis was IC3.

PIN Set Relation {Argument Set}

This selection variable is similar to DEVICE, in that it refers to failures detected by the guided probe, but this set refers to pins rather than just IC's. The pins may be externals ('CLOCK'), a single IC pin ('IC5.8'), or a range of pins on one IC ('IC5.8 TO .10'). As an example,

PIN INTERSECTS {EXT3, IC4.2, IC6.9 TO .11}

would be true if the guided probe diagnosis included EXT3, IC4.2, IC6.9, IC6.10, or IC6.11.

COMMENT CONTAINS 'string'

The final relational operator is the substring comparison, CONTAINS. It is used to test portions of the operator's comment, if one was entered. A 'string' may contain '@', which acts as a match-anything character. For example,

COMMENT CONTAINS '#1792-@@@@'

would be TRUE if the operator comment contained the substring '#1792-' followed by any four characters.

Item Relation

There are two other relations available, both bases on the presence of items in the current record.

Item PRESENT

This relation is TRUE if the specified item was enabled for logging and is present in the current record. All 'items' are listed in Table C-1.

Item ABSENT

This relation is TRUE if the specified item was not enabled for logging, or if it was enabled but is not present in the current record. (Equivalent to NOT item PRESENT).

Examples:

```
IF DATE PRESENT THEN PRINT DATE;

IF FFT ABSENT THEN PRINT 'NO FFT';
```

BOOLEAN OPERATORS

The Boolean values derived from relational operators may be combined into more complex expressions using parentheses and the Boolean operators AND, OR, and NOT. For example, the Boolean expression

```
FFT >= 1000 AND FFT < 2000
```

is TRUE for first failing test numbers between 1000 and 1999.

Output Formats

With the exception of literal strings, which are output as they appear, each item is printed in a fixed format to facilitate further processing. Those formats and the interpretation of the items follow. Some FORTRAN format notation is used. The FORTRAN format notation that is used is quite elementary and is as follows:

I=Integer

X=Space

A=Alpha-numeric character

In: where n is the number of digits in the integer field.

The field is right justified and leading zeros are suppressed.

nX: where n indicates a particular number of spaces

nAm: where n indicates the length of alpha-numeric field and m is always one. The alpha-numeric field is left justified and filled with blanks.

n(y,x): where the information inclosed within the parentheses is repeated n times per line.

All items (but not literal strings) end with a carriage return and line feed. If an item is not present in a record, a default item is printed. (See the end of this section).

NOTE

> In the following examples, spaces that are not indicated by a slashed b (b̸) do not exist, but are in the examples to make them more readable.

DATE

This item is printed as MM/DD/YY. For example:

3/09/79

FFT

This is the first failing test step number, output in an I6 format.

The I6 format indicates that the FFT must be an integer and has a maximum field width of six. For example:

```
ƀƀƀƀƀ8
```

indicates that the FFT is eight.

DIAGNOSIS

For AFL diagnoses or for guided probe diagnoses with DIAGNOSIS not enabled, the diagnosis source is printed in I3 format. The full guided probe diagnosis is output as: diagnosis (I3), 2X, number of following lines (I3). The succeeding lines might appear as:

```
          IC3ƀƀƀƀƀ . ƀ4 ƀƀƀ 0010

          CLOCKƀƀƀ ƀ ƀƀ ƀƀƀ 1100

FORMAT      8A1      I2 3X  4I1
```

The four digits at the end specify the following conditions on the pin:

digit 1	stuck	0 no
		1 yes
digit 2	level	0 no error
		1 bad-high
		2 bad-low
		3 bad-mid range
digit 3	pulse	0 no error
		1 missing
		2 illegal
		3 oscillation
digit 4	half pulse	0 no error
		1 illegal

HISTORY

This is a compressed version of the probe history file. The first line is the number of probe points (I3). One line for each point follows. Example lines might be:

```
          23Eƀƀƀƀƀ ƀ ƀƀ ƀƀƀ  2  ƀƀƀ ƀƀƀƀƀ1 ƀƀƀ ƀ8

          IC5ƀƀƀƀƀ . ƀ4 ƀƀƀ  6  ƀƀƀ ƀƀƀƀƀ1 ƀƀƀ ƀ2

FORMAT       8A1      I2 3X  I1  3X    I6   3X  I2
```

which contain the pin probed, the pin condition, first failing test for that probe point, and message number.

Pin conditions are:

1	good
2	bad
3	pass
4	--
5	good (conditional)
6	bad (conditional)
7	pass (conditional)

Messages are:

0	none
1	BACKUP
2	CHECKPOINT
3	DIRECT (U)
4	FINAL
5	JUMP-TO
6	SELF-PROBE
7	START (A)
8	START (P)
9	START (U)
10	TENTATIVE

COMMENT

This item is printed in the same form as it was entered, except that '@' entered on the TEC scope is converted to space. For example:

```
                    BOARD SERIAL NUMBER 12
```

FFTV

This item is output as the number of failing driver/sensors (I3), followed by successive lines of failing driver/sensor numbers. The format of these succeeding lines is I3, 9(1X, I3). For example:

```
                    ƀƀ1 ƀ ƀƀ2 ƀƀ3 ƀ ƀƀ4 ƀ ƀƀ5 ƀ ƀƀ6 (for a maximum of ten entries
                                                   per line)
FORMAT              I3            9(1X,I3)
```

FBTV

This item is output as the number of "runs" which follow (I4). A "run" is defined as a failing test and the number of tests immediately after which also fail. For example, 1034(10) represents the fact that test 1034 and the 10 steps following it failed. Each run is printed as I6, '(', I3, ')'. Each line is output as:RUN,4(2X.RUN).

For example:

b̸b̸b̸9

FORMAT I4

FORMAT $\underbrace{\text{b̸b̸1012}}_{\text{I6}}\ \underbrace{\text{(b̸b̸3)}}_{\text{I3}}\ \underbrace{\text{b̸b̸}}_{\text{2X}}\ \underbrace{\text{b̸b̸1120}}_{\text{I6}}\ \underbrace{\text{(b̸b̸6)}}_{\text{I3}}$ (for a maximum of five entries per line)

CLASS

This item is output as the number of fault classes which follow (I4). Succeeding lines are output as I6, 7(2X, I6). For example:

FORMAT $\underbrace{\text{b̸b̸b̸b̸b̸1}}_{\text{I6}}\ \underbrace{\text{b̸b̸}}_{\text{2X}}\ \underbrace{\text{b̸b̸b̸b̸b̸7}}_{\text{I6}}\ \underbrace{\text{b̸b̸}}_{\text{2X}}\ \underbrace{\text{b̸b̸b̸b̸b̸8}}_{\text{I6}}$ (for a maximum of eight entries per line)

The default outputs for an item appearing in a PRINT statement which is not present in the current record are:

DATE	00/00/00
FFT	000000
DIAGNOSIS	no default needed
PROBE	000
COMMENT	blank line
FFTV	000
FBTV	0000
CLASS	0000

Input File

The input file for the report generator is an ASCII file, prepared with the system editor. The language is free-form: spaces and other formatting characters are only required to the extent that they separate two adjacent symbols in the language. Users are encouraged to make use of this fact and indent blocks and parts of conditionals as in some of the examples. Keywords may be abbreviated to the shortest recognizable contraction (such as SUP for SUPERSET). However, this practice is discouraged, as future changes to the report generator language may invalidate the contractions. Comments are permitted anywhere except in the middle of keywords and numbers. They consist of any characters bracketed by ' ', as in

```
                    THIS IS A TWO-

                  LINE COMMENT
```

Input Errors

Errors encountered while reading the user's program will be reported on the Teletype and processing of the log file will be aborted. An EXECUTION SUPPRESSED message will be printed. Errors the user might make fall into two classes: syntactic errors and specification errors. The former are typified by misspelled keywords, unbalanced parentheses and BEGIN...FINISHes. The latter occur when specifying non-existent EXTERNAL or IC names or statement labels.

The format of these error messages is the same. The first line gives the error type and the line and page numbers at which the error occurred. The following line or lines give the context of the error. '???' marks the place where the error was detected. The first character of the error message gives its type. '!' marks syntactic errors and '%' marks specification errors. Some errors may cause others to be overlooked on a particular report generator run. As a result, correcting all the errors detected by the report generator one time may not permit a successful run the second time, although it usually does.

Examples:

```
! SYNTAX ERROR AT LINE 1, PAGE 1

IF DATE ABSENT ??? PRINT

% ILLEGAL DATE AT LINE 2, PAGE 1

IF DATE = 3  / 9  / ??? 1979
```

C.9 CAUTIONS

The diagnosis logging facility is quite sensitive to changes in the .SR and .TP files. Rerunning Prepare Mode will make it illegal to log further information in an old log file. However, the information contained therein will be valid. Attempts to print the log at this point will generally produce the message:

```
WARNING:   .DL/.LT MISMATCH MAY CAUSE ERRONEOUS ITEM SELECTION.
```

Attempts to select CAPS steps using a statement label (e.g., 10:) or 'LAST' may produce the erroneous results mentioned.

If logging is attempted during Diagnosis Mode, and it is determined that the .SR or .TP information has changed, which would make it impossible to log the new data, the following hard error occurs:

```
PROGRAM FILES HAVE CHANGED.

CAN'T ADD MORE DIAGNOSES TO dev:netwrk.DL
```

The best procedure to follow when planning to make changes to the .SR or .TP files is:

1. Before running Prepare Mode, run the log report generator to obtain all saved information.

2. Make desired .TP and/or .SR changes, run Prepare Mode.

3. Before running Diagnosis Mode, initialize a new log file.

C.10 LOG MODE PRINTOUT

Typical LOG mode printouts are illustrated in Figure C-8. The printouts illustrated are DATE, DIAGNOSIS, COMMENT, PROBE HISTORY, FIRST FAILING TEST VECTOR, FIRST FAILING TEST, and FAIL BY TEST VECTOR.

```
/


/   *** DATE ***
 3/15/77


/   *** DIAGNOSIS ***
  1                  /1 = # OF ENTRIES INDIAGNOSTIC LIST
105                  /105=NODE FAILURE
IC2    8  0200       /IC2. 8 ACTIVE LOW


/   *** COMMENT ***
THIS IS A COMMENT !!!!!!!!!!


/   *** PROBE HISTORY ***
  5                        /5= # OF ENTRIES IN HISTORY LIST
22E       2      1   8     /22E BAD @ #1 START(P)
IC2    8  6      1   2     /IC2. 8 BAD(COND) @ #1 CHECKPOINT
IC2    9  7     36   0     /IC2. 9 PASS(COND) @ #36
IC2    9  7     36  10     /IC2. 9 PSS(COND) @ #36 CONDITIONAL
22E       2      1   6     /22E BAD @ #1 SELF PROBE


/   *** FIRST FAILING TEST VECTOR ***
  1                                /# OF ENTRIES
 34                                /# OF FAILING DRIVER-SENSORS


/   *** FIRST FAILING TEST ***
    1                              /FIRST FAILED AT TEST # 1


/   *** FAIL BY TEST VECTOR ***
   4                               /# OF ENTRIES
    2( 45)     49( 51)    114(  0)     116( 30)
/2( 45)=FAILED TEST 2 AND THE FOLLOWING 45 TESTS
```

Figure C-8. LOG Mode Printouts

Nodal Verification – Appendix D

D.1 INTRODUCTION

D.1.1 General

Even when a test program with an acceptable percentage completion has passed a "Known-Good-Board", there is the possibility that the simulation values of nodes internal to the board might not truly reflect the values those nodes really exhibited. Therefore, if these mismatched nodes were ever probed while trying to find a fault on a bad board, the probe would probably diagnose an error at that point, not because an error really existed, but because the actual board values and simulated values differed.

There are several reasons for these mismatches:

a. Network description errors.
b. Test Program races.
c. Incomplete test programs.
d. Incorrect initialization sequences.
e. Incorrect IC Models.

Nodal Verification allows the user to probe every node on the "Known-Good-Board" in an orderly sequence and without any modification to the test program.

Nodal Verification can be used, to automatically create a Pulse Discrimination (.UD) file. This .UD file is used in the DIAGNOSE Mode to resolve conflicts between the UUT and the simulation data base caused by inconsequential pulses that may be detected by the sensitive GenRad pulse-catching probe. Nodal Verification automatically sets up a list of points to be probed (based on the data contained in the Network (.SR) File) and collects all data on illegal, missing, and half-pulses from the probed UUT. This data is used to create an IGNORE type .UD file that determines which test steps to ignore.

Nodal Verification depends on probed values from a "good UUT" and, realizing that there may be differences between several UUT's, it may be necessary to verify the resulting data base by probing several UUTs. Following an initial NODALV operation, an operator can probe another UUT and compare the results against the previous NODALV data base. Any differences between the two data bases can be listed, and a new .UD created to incorporate the differences. If no differences are found then the validity of the NODALV created .UD file has been established.

Nodal Verification probing sequences can be stopped at any time and restarted at a later date with no loss of accumulated data. The probing sequence will return to the next unprobed point; this will prove useful for boards containing large numbers of ICs. In addition, the probing sequence and data translation can be run separately. Once a probing sequence has been performed, the operator can remove the mass storage device from the tester and run the data translation on a programming station, freeing the tester for other operation.

D.1.2 NODALV Option page

Nodal Verification is supplied as a seperate mode of the CAPS/APG Monitors. Prior to running Nodal Verification the operator must have first run the PREPARE mode with the N,T,G, and D options turned on. To enter Nodal Verification, the operator then types NODALV (XMIT). The option page listed in Figure D-1 will be displayed:

```
---------------------------------------------------------------------------
#                                                          NODAL VERIFICATION
        FILE USAGE

N  = DSK:BDNAME.SR      NETWORK FILE
UR = DSK:NONE.UR        USER SUBROUTINE FILE
LI = LPT:               LISTING DEVICE FOR NVTRAN
W  = SYS:, DSK:         WORKING DISKS

        OPTIONS

FT = NO    FIRST TIME PROBING THIS BOARD (YES/NO)
M  = BOTH  MODE - PROBE, DIFFERENCES, BOTH P&D,SHORTS CHECK
I5 = NO    USE TRANSLATED VERSION OF OLD .UD TO DETERMINE DIFFERENCES (YES/NO)
UD = YES   MAKE A NEW .UD IF DIFFERENCES FOUND (YES/NO)
AD = NO    ADD TO OLD .UD MADE BY NVTRAN WHEN MAKING .UD (YES/NO)
C  = NO    USE CLIP (YES/NO)

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO THIS MODE
---------------------------------------------------------------------------
```

Figure D-1. NODALV Options Page.

The option lines are defined as follows:

N = Same as the Network (.SR) file used in the PREPARE Mode.

UR = Same as the User subroutine file used in the DIAGNOSE Mode (assembly language subroutines).

LI = The listing device for NVTRAN data outputs.

W = Working surfaces.

FT = The first time the board is probed, this option must be set to YES. This option creates the temporary files that will store probe data (Verification Path .VP, and Verification Data .VD). Following each run, the temporary files are updated. To reprobe a board set the option to NO.

M = The Mode consists of four options, PROBE, DIFFERENCES, BOTH Probe & Differences, and SHORTS CHECK.

PROBE, enters probe data into the .VP and .VD files and, following the probing sequence, returns to the system monitor without translation.

DIFFERENCES, compares the results of the probing sequence against the Good Board Simulation (.GV file). Differences between the simulation and probing files are then output to the listing device. If probing has not been completed, the translation will contain only those points probed. If probing has not been performed an error message results.

BOTH Probe & Differences, probe the UUT and enter the probing data into the .VD and .VP files. Translates the data, and compare the results against the simulation.

SHORTS CHECK, This option overrides all other option and allows clipping of all ICs on a board to check for near pin opens and shorts. This allos the operator to check the board for shorts prior to using FINDS. Refer to Para. D.5 for SHORTS CHECK operation.

I5 = Use the translated version of the old .UD to determine the differences between the old .UD and the present probing sequence. Put the differences found (if any) in the .DN (Differences) file. The I5 and UD options should normally not be ON at the same time, unless AD is also ON.

UD = Create a new .UD if the resulting probing data is different than the previous probing data. Delete the previous .UD and make a new file using the new probe data.

AD = If UD is on, add new probe data to the established .UD.

C = For probing using the clip only. The resulting data cannot be translated.

D.2 USING NODALV

D.2.1 General

Nodal Verification consists of a UUT probing sequence that uses the GR Pulse-Catching Logic Probe to establish a pulse discrimination data base. This data base is then translated by the Nodal Verification Data TRANslator (NVTRAN) to produce a Pulse Discrimination (.UD) file for use in the DIAGNOSE Mode, and a Difference file, .DN (if any), for verification of the existing .UD.

The IC Clip and X1 Probe can be used to determine bad levels but the resulting data files cannot be translated to create a .UD since the probing data is incomplete (no pulse discrimination).

D.2.2 GenRad-Pulse Catching Probe, X1 Probe, and IC Clip

Nodal Verification is most powerful when used with the GenRad Pulse-Catching Probe, however, added flexibility is provided by use of the IC Clip feature. To prepare complete Pulse Discrimination files the probing should be performed by using the GenRad Pulse-Catching Probe. For quick check verification of bad levels, the Clip provides the operator with the ability to rapidily test large numbers of ICs. The X1 probe provides the bad level check capability but does not detect pulses.

The Pulse-Catching Probe and Clip can be used together in probing sequences but the resulting .UD will have limitations since the Clip does not provide any pulse catching. To switch from the pulse-catching Probe to the Clip during a probing sequence, the operator simply depresses the (XMIT) button and the display is updated to indicate the type of probing being performed.

D.2.3 Probing

As stated previously, UUT's are probed in Nodal Verification by use of the GR Pulse Catching, X1 Probe, or IC Clip. The purpose of the

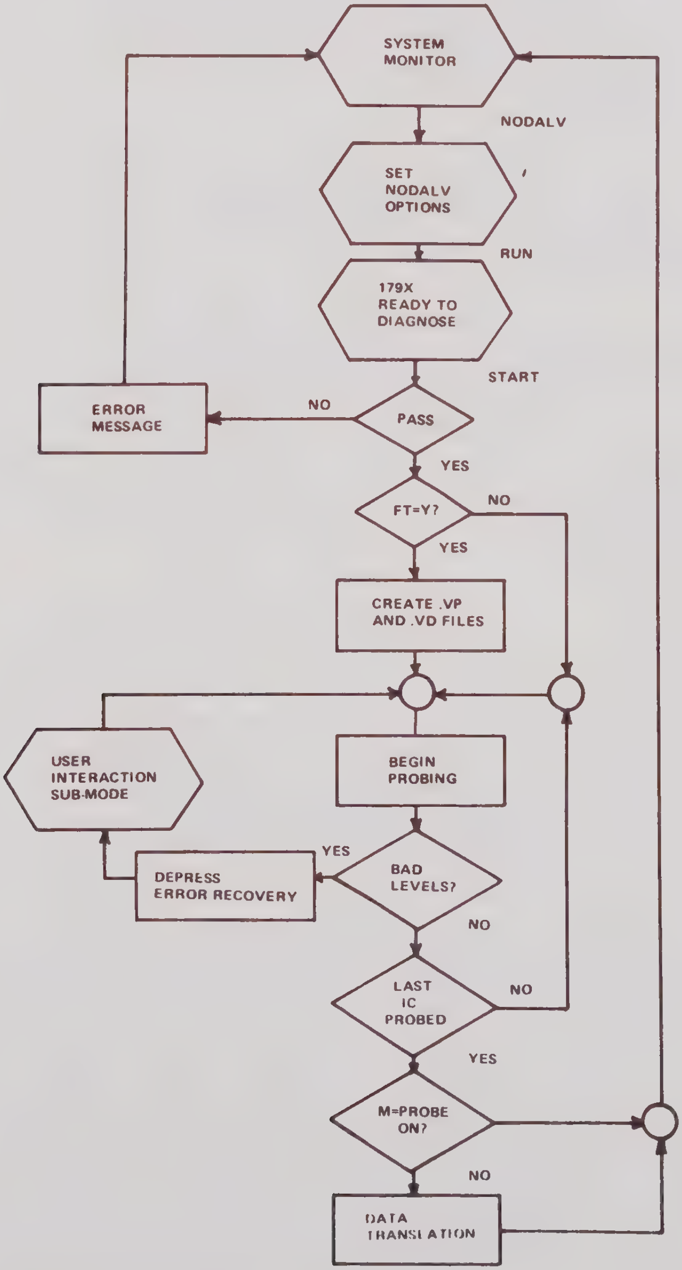

SYSTEM MONITOR
NODALV
SET NODALV OPTIONS
RUN
179X READY TO DIAGNOSE
START
ERROR MESSAGE
NO
PASS
YES
FT=Y?
NO
YES
CREATE .VP AND .VD FILES
USER INTERACTION SUB-MODE
BEGIN PROBING
YES
DEPRESS ERROR RECOVERY
BAD LEVELS?
NO
LAST IC PROBED
NO
YES
M=PROBE ON?
NO
DATA TRANSLATION

Figure D-2. Probing Sequence.

probing sequences is to establish or add to two data base files; the Verification Path (.VP) and the Verification Data (.VD)files.

Figure D-2 is provided as an example of the probing sequences. To perform an initial probing sequence on a UUT the NODALV options must be set as follows:

```
FT =    NO    FIRST TIME PROBING THIS BOARD (YES/NO)
M  =    BOTH  MODE - PROBE, DIFFERENCES, BOTH P&D,SHORTS CHECK
```

Setting FT = YES creates the .VD and .VP files which have the same name as that established for the Network file. Setting M = BOTH will result in the data translation of the probing data base into the .UD following the probing sequence. M = could have been set to PROBE if translation was not required immediately. Data base files are still established, however.

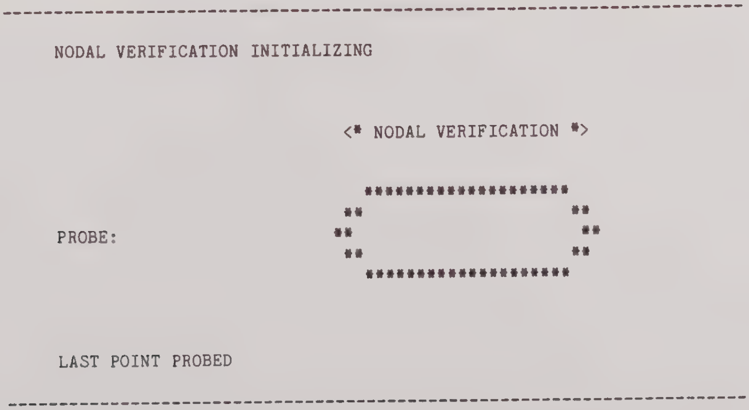
Refer to Figure D-2. The operator enters the command RUN (XMIT) and the system responds by initially diagnosing the board to assure proper operation. If the UUT is verified "good", the system initializes NODALV (initializing data base files) prior to beginning the probing sequence. If the UUT fails, the system returns to the system monitor and prints the following message on the TTY.

```
BOARD FAILURE IN NODAL VERIFICATION - USE DIAGNOSE MODE

ERROR #            AC#
```

Nodal Verification cannot be performed on this UUT until the failure is diagnosed and repaired.

Once the initial diagnosis is successfully completed the system enters the probing sequence. Following the RUN (XMIT) command, the system displays the 179X READY TO DIAGNOSE message. Depress the START button to perform the initialization.

Following the successful intialization, the START Lamp will light. To continue, depress the START button. The .VP and .VD files will be initialized prior to the probing sequence, and all points specified in the .SR file will be probed. While the system is initalizing the .VD and .VP files, the Nodal Verification Initializing page is displayed as shown in Figure D-3.



```
NODAL VERIFICATION INITIALIZING


                         <* NODAL VERIFICATION *>

                          ********************
                        **                    **
PROBE:                 **                      **
                        **                    **
                          ********************


LAST POINT PROBED
```

Figure D-3. Nodal Verification Initializing Message.

Once the files have been initialized, the NODALV "working" page is displayed, the PROBE lamp lights, and the TTY bell rings. The NODALV probing can consist of a probing or clipping sequence, or a combination of the two.

NOTE

> Compilation of NODALV data bases by use of the clip alone will not be translated by NVTRAN. The result of Clip data will be a listing of bad levels.

The NODALV working page displays the first point to be probed as shown in Figure D-4.

```
------------------------------------------------------------------------

    MESSAGE:


                                   <* NODAL VERIFICATION *>

                                     ********************
                                   **                    **
    PROBE:   (14 PINS)            **     C1.2              **
                                   **                    **
                                     ********************


    LAST POINT PROBED: C1.1
------------------------------------------------------------------------
```

Figure D-4. NODALV Probing Sequence.

The operator now probes the designated point and depresses the PROBE button. When the data transfer has been completed, the screen will be updated with the next probing point, the PROBE lamp lights, and the TTY bell rings.

When the last IC has been probed, the NVTRAN begins to translate the data files and, when completed, returns to the system monitor.

If using the Clip, C = YES, the clip data files can not be translated by NVTRAN. The probing sequence displays are slightly altered during clipping. The operator then attaches the IC Clip to the designated IC. Note, that only the IC number is indicated not the pin number. When the operator depresses the PROBE button, the LAST POINT PROBED: message will be incremented displaying each IC pin number as it is tested, for example:

```
LAST POINT PROBED: C1.1
```

updates to read

```
LAST POINT PROBED: C1.3
```

then

```
LAST POINT PROBED: C1.5
```

and so on until the last IC pin is displayed. At this point the working page displays the next IC and rings the TTY bell awaiting the next clipping.

The operator then clips C2 to continue. Following the clip sequence the system returns to the system monitor.

D.2.4 User-Interaction Sub-Mode

When a bad level is detected while performing NODALV probing, the operator may reprobe the faulty pin or IC by use of the User-Interaction Sub-mode. When the bad level is detected, the system displays the message; BAD LEVEL FOUND ON THIS NODE

If the operator wants to reprobe the point, he depresses the ERROR RECOVERY button and the system enters the sub-mode. The User-Interaction Sub-mode contains five options which are displayed on the option page as shown in Figure D-5.

```
------------------------------------------------------------------------

             WAITING FOR INPUT

                           <* NODAL VERIFICATION *>

                                ********************
                              **                    **
  PROBE:                     **                      **
                              **                    **
                                ********************


                     USER-INTERACTION SUB-MODE

                     A=RETURN TO AUTOMATIC SUB-MODE
                     E=EXIT NODAL VERIFICATION
                     R=REPROBE THE LAST POINT OR RECLIP THE LAST IC
                     #=USER DEFINED IC. PIN #
                     *=DELETE CLIP DATA FOR IC


     LAST POINT PROBED: C2.8

------------------------------------------------------------------------
```

Figure D-5. User-Interactive Sub-Mode Display.

The operator now has five options, he can: Reprobe the last point probed (R), return to the normal probing sequence (A), exit from Nodal Verification (E), specify a point to be reprobed (#), and delete IC Clip data (*).

If the operator wishes to reprobe the point where the bad level was detected (the last point probed), enter R (XMIT). The TEC screen is updated to indicate the point to be probed. The operator then reprobes and depresses the PROBE button. The previous probe data for the pin is deleted and the new probing data entered into the data base file. To continue the reprobing, the operator enters the command A (XMIT) to exit from the User-Interaction Sub-Mode, and NODALV continues on in the normal probing sequence. If the bad level is valid, and not caused by improper clipping or bad contact, the operator must determine whether to continue the probing sequence or exit from Nodal Verifiication.

If the operator wishes to specify the point to be probed, he enters the sub-mode and enters the IC specification by use of the sharp (#) command, as for example, to specify IC2 pin 1 enter:

#C2.1 (XMIT)

The system responds by redisplaying the sub-mode page with the message ARE YOU SURE?, allowing the operator to reconfirm the entry. To continue the operator enters a Y (XMIT) and the system displays the updated option page.

The operator then reprobes. To exit from the sub-mode enter A (XMIT) and the system returns to the last point probed previously.

If the operator selects a point identified as being on the same node as another IC probed previously, NODALV will print the message - THIS PIN IS ON THE SAME NODE - and then overrides the IC specification and indicates the other IC as the point to be probed. When using the User-Interaction Sub-Mode with the IC Clip, the system will only reprobe the last pin probed on the IC. To reprobe the entire IC the operator must use the asterisk (*) command to delete the previous probing data for the IC and then reclip.

D.3 NVTRAN

Nodal Verification data base files (.VD and .VP) are created during the probing sequence and then used in NVTRAN to create the Pulse Discrimination file (.UD). The .UD is then translated for use by the test system. NVTRAN is automatically entered following the probing of the last IC. The translation is performed without any user interaction. NVTRAN is enabled whenever M=BOTH or DIFF, as long as no bad levels are detected, and, the data base files do not consist of Clip-only data (C=YES).

The flow sequence for NVTRAN (refer to Figure D-6) is included in this discussion for general information only. The initial step in NVTRAN is to determine whether the I5 option is ON or OFF. If I5 =YES, NVTRAN will compare the probed data base with the good simulation data base, and use the existing .I5 file to resolve any pulse conflicts. If there are no conflicts and the .UD is not an IGNORE ALL type .UD, NVTRAN will create a differences file (.DN) that contains the message, "CONGRATULATIONS - NO NODAL DIFFERENCES" and will output this file to the listing device. In addition, the same message is sent to the TTY. Following the data output, NVTRAN returns to the System Monitor.

If I5=NO, NVTRAN will compare the probing data base against the good simulation data base, .GV. If there are no differences and the Clip (or Clip and Probe) are used during the probing sequence NVTRAN will output the message "NO LEVEL DIFFERENCES" to the TTY. If the data base files consist entirely of GenRad pulse catching probe data and no differences are found, the .DN contains the message "NO NODAL DIFFERENCES", and the file is output to the listing device while the same message is sent to the TTY.

If nodal differences are found with either I5 ON or OFF and UD=YES, NVTRAN prepares to create a new .UD, and if AD=YES, add to an existing .UD. NVTRAN first verifies that no bad levels exist, if any do and UD=YES the message "BAD LEVELS FOUND!! - UNABLE TO MAKE OR ADD TO .UD" is sent to the TTY. If UD=NO the bad levels are put in the .DN and the file outputted to the listing device.

If no bad levels were found, NVTRAN checks to see whether the operator wishes to make a new .UD (UD=YES, AD=NO) or add to the existing one (AD=YES). If creating a new .UD, NVTRAN makes an IGNORE type pulse discrimination file and runs it through the Discriminator for translation. Once translated the new .UD and the .DN file are output to the listing device and the system returns to the system monitor.

If adding to the existing .UD (AD=YES), NVTRAN will make a .UD for all nodes with pulse differences and all nodes which have entries in the old .UD. NVTRAN then translates the data bases and lists the .DN and the updated .UD on the listing device before returning to the system monitor.

D.4 ERROR MESSAGES

NVTRAN provides two types of error messages explained as follows:

Good Messages:

"CONGRATULATIONS - NO NODAL DIFFERENCES" - Indicates that a successful probing sequence has been completed, and the resulting data files, when compared against the good board simulation (using the user discrimination file .I5), found no nodal differences.

"CONGRATULATIONS - NO LEVEL DIFFERENCES"

Indicates that the board has been clipped successfully and all bad levels detected. NVTRAN is unable to provide a .UD using data files consisting of Clip-only data, therefore, if the operator wishes to make a .UD, the board must now be probed with the GR Logic Probe.

Bad Messages:

"RAN OUT OF ROOM MAKING THE .UD" - Indicates insufficient disk space to create the .UD file.

"UNABLE TO ADD TO ENABLE TYPE .UD" - NVTRAN can use an ENABLE type .UD to find nodal differences, but cannot make an ENABLE .UD or add to an existing ENABLE .UD.

"CURRENT .I5 IGNORES ALL PULSE CATCHING" - Indicates that the present User Discrimination Data Base file (.I5) is an IGNORE ALL type file. An IGNORE ALL file ignores all pulse catching.

BAD LEVELS FOUND !! - UNABLE TO MAKE OR ADD TO .UD" - The board must green light internally before creating the .UD file otherwise, incorrect diagnosis could occur.

"WARNING PULSE DISCRIMINATION INCOMPLETE - SUGGEST YOU PROBE ENTIRE BOARD" - Indicates that the UUT was not completely probed and that a resulting .UD will be incomplete.

"NO GR LOGIC PROBE DATA! CAN'T MAKE A .UD" - Indicates that the data base files consist of Clip-only data which cannot be used by NVTRAN.

D.5 SHORTS CHECK MODE

The purpose of the Shorts Check Mode is to allow an operator to quickly clip an entire circuit board to find near pin opens and shorts. The Shorts Check would normally be performed prior to using FINDS in DIAGNOSE. This quick check capability is intended to minimize bad diagnoses, and to verify the connectivity specified by the .SR file.

To run Shorts Check, specify the files, listing device, and working device; then set the M= option to SHORTS. This command overrides all other option commands so that it is not necessary to reset FT=, or to enable the clip (C=YES), the necessary options are enabled automatically. Type RUN and the working page is displayed indicating the first IC to be clipped.

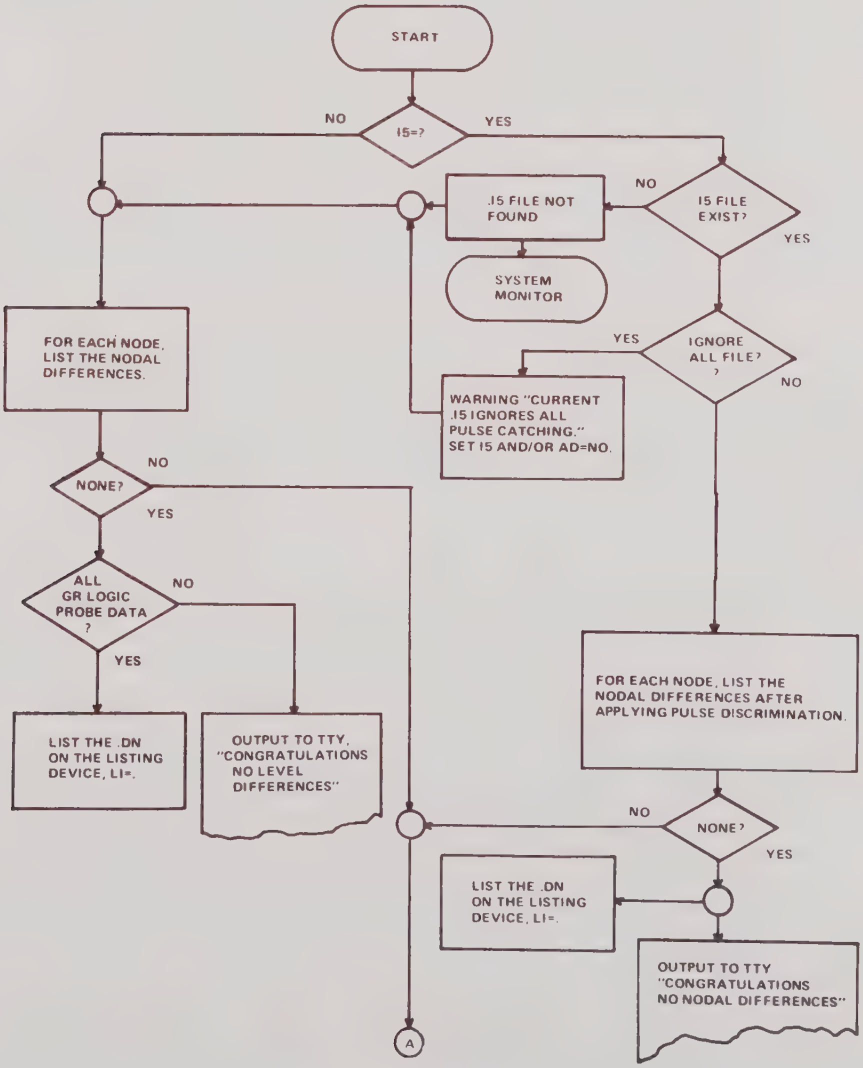

START
I5=?
NO
YES
.I5 FILE NOT FOUND
NO
I5 FILE EXIST?
YES
SYSTEM MONITOR
FOR EACH NODE, LIST THE NODAL DIFFERENCES.
YES
IGNORE ALL FILE? ?
NO
WARNING "CURRENT .I5 IGNORES ALL PULSE CATCHING." SET I5 AND/OR AD=NO.
NONE?
NO
YES
ALL GR LOGIC PROBE DATA ?
NO
YES
LIST THE .DN ON THE LISTING DEVICE, LI=.
OUTPUT TO TTY, "CONGRATULATIONS NO LEVEL DIFFERENCES"
FOR EACH NODE, LIST THE NODAL DIFFERENCES AFTER APPLYING PULSE DISCRIMINATION.
NO
NONE?
YES
LIST THE .DN ON THE LISTING DEVICE, LI=.
OUTPUT TO TTY "CONGRATULATIONS NO NODAL DIFFERENCES"
A

Figure D-6. NVTRAN Flow.

The operator responds by clipping the appropriate IC and depressing the PROBE button. The next IC to be clipped is then displayed. When a failing point is found, the system responds with the message WORKING ON FINAL DIAGNOSIS, and waits for the operator to depress the PROBE button once again. The diagnostic message is then displayed, and is in the FINDS Diagostic message format as shown in Figure 6-11. The DIAGNOSIS FROM: will read INTERMEDIATE FINDS.

The Shorts Check will stop at the first failing point, diagnose the failure and then return to the System Monitor. To continue verifying the UUT, the operator can return to NODALV, RUN Shorts Check, and by continuously depressing the XMIT button, return to the failing point. The operator can then depress the XMIT button again to skip over the failing IC and then continue clipping the rest of the UUT.

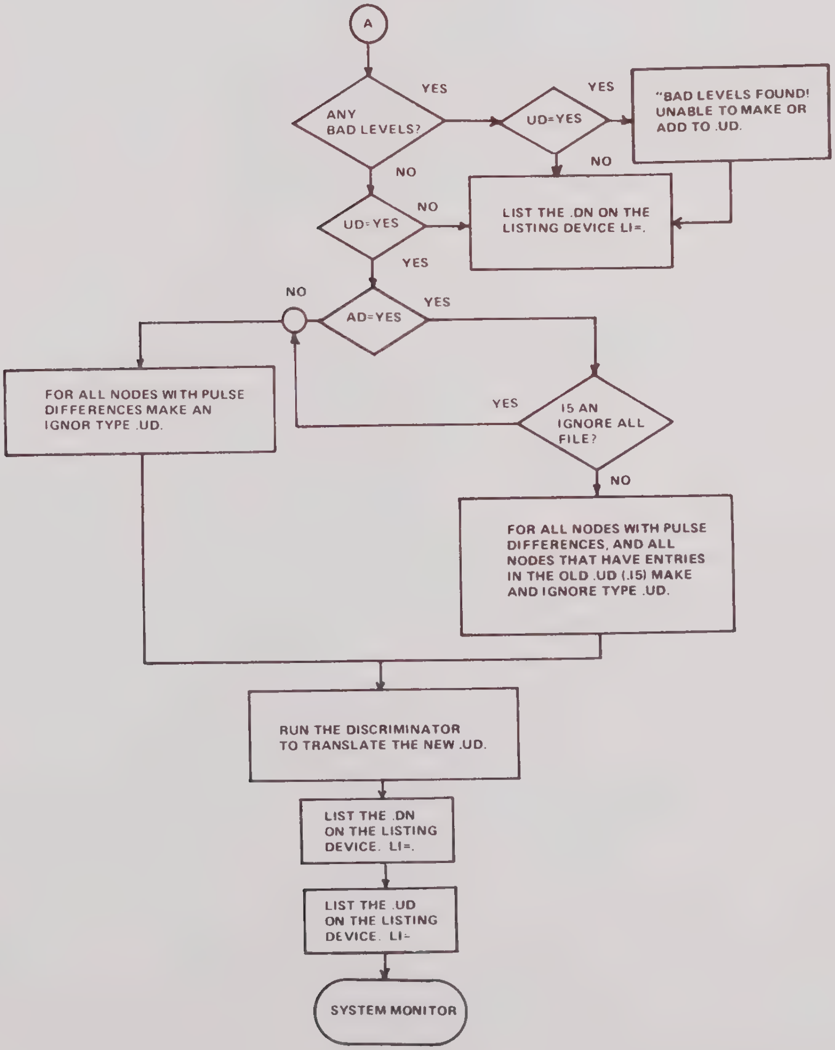

A
ANY BAD LEVELS?
YES
NO
UD=YES
YES
NO
"BAD LEVELS FOUND! UNABLE TO MAKE OR ADD TO .UD.
LIST THE .DN ON THE LISTING DEVICE LI=.
UD=YES
NO
YES
AD=YES
NO
YES
FOR ALL NODES WITH PULSE DIFFERENCES MAKE AN IGNOR TYPE .UD.
IS AN IGNORE ALL FILE?
YES
NO
FOR ALL NODES WITH PULSE DIFFERENCES, AND ALL NODES THAT HAVE ENTRIES IN THE OLD .UD (.15) MAKE AND IGNORE TYPE .UD.
RUN THE DISCRIMINATOR TO TRANSLATE THE NEW .UD.
LIST THE .DN ON THE LISTING DEVICE. LI=.
LIST THE .UD ON THE LISTING DEVICE. LI=
SYSTEM MONITOR

Figure D-6. Continued.

2225 Mode–Appendix E

E.1 INTRODUCTION

This appendix contains procedures for converting board-testing files prepared with CAPS/APG software on a PDP-8 based test system to files suitable for use by a GenRad 2225 Portable Service Tester. Additionally, the method for transfer of any appropriate file to the 2225 from a PDP-8 based system is described.

Before any transfer of testing files to a 2225 across the PDP-8 link can take place, certain CAPS files must be translated successfully by two 2225 Mode translators. One of these translators, referred to as the Image Translator, converts a CAPS data-base file (.SH) obtained by running a network description, into a 2225 image file (.IM). The other 2225 Mode translator is referred to as the Test Program Translator that converts a system test program designed to use simulator outputs (normally the .DA or .AP auto program output file), for use by the 2225. The Test Program Translator output file has a .T2 extension. Both the .IM and .T2 files are required for board testing on the 2225. Before either translator can be used however, an "equate" file (.EQ) that correlates system driver-sensor pins with 2225 driver-sensor pins must be available. Also, the CAPS and user libraries must be available to develop .IM testing files.

E.2 REQUISITES

To ready a 2225 for board testing using data-base files developed on a test system using CAPS/APG software requires that certain PREPARE sub-modes be run, and 4 CAPS libraries available as described in the following paragraphs. Figure E-1 illustrates the 2225 Mode flow.

Network Description File (.SH) and Image Translator

The PREPARE N, T, and G submodes are run to take a network description .SR file and process it to obtain the .SH file. The Image Mode translator converts the .SH file to a file with a .IM extension that can be used by the 2225. The name of the network (.SH) file to be translated is specified by the N= file usage option as shown in Figure E-2. The Image translator is selected with the 2225 Execute I sub-mode (Figure E-2). To convert the .SH file, the I translator requires an equate file as described below.

During the processing of the .SH file, a Warnings file may be created as described in para E.7. This file warns the user of any irregularities encountered by the translator.

Test Program Translator

The 2225 Mode test program translator converts a test program and creates a testing file with a .T2 extension. Though the test program file to be translated can be any test program source, it is recommended that the .DA file be used because the simulated digital output patterns are included in the program, resulting in a functional program. Prepare sub-modes N, T, G, and D must be run to obtain the .DA file. The Test Program Translator is selected with the 2225 Execute T sub-mode (Figure E-2). To convert the test program, the Image Translator requires an equate file as described below.

Any commands that cannot be translated because they have no 2225 equivalent are placed in the Translator Warnings (.TW) file.

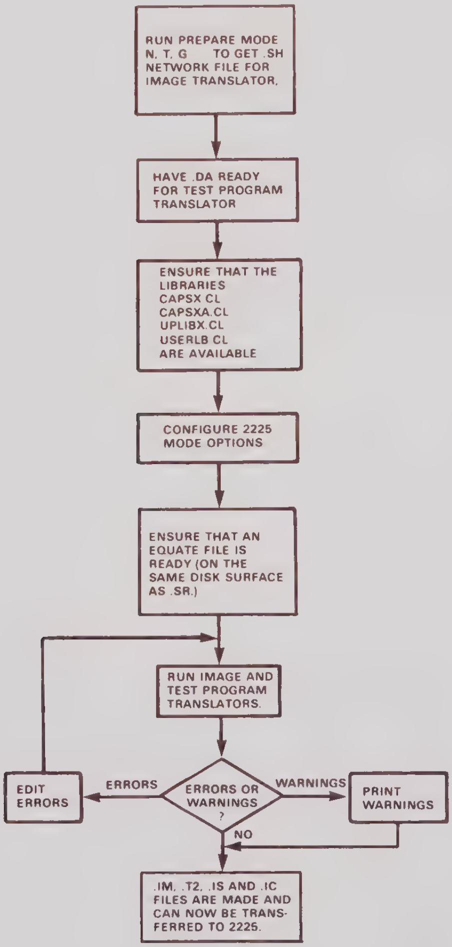

RUN PREPARE MODE N, T, G TO GET .SH NETWORK FILE FOR IMAGE TRANSLATOR.
HAVE .DA READY FOR TEST PROGRAM TRANSLATOR
ENSURE THAT THE LIBRARIES CAPSX.CL CAPSXA.CL UPLIBX.CL USERLB.CL ARE AVAILABLE
CONFIGURE 2225 MODE OPTIONS
ENSURE THAT AN EQUATE FILE IS READY (ON THE SAME DISK SURFACE AS .SR.)
RUN IMAGE AND TEST PROGRAM TRANSLATORS.
EDIT ERRORS
ERRORS
ERRORS OR WARNINGS ?
WARNINGS
PRINT WARNINGS
NO
.IM, .T2, .IS AND .IC FILES ARE MADE AND CAN NOW BE TRANSFERRED TO 2225.

Figure E-1. 2225 Mode Flow.

Equate File (.EQ)

An "equate" file with the extension .EQ must be made to correlate test system driver sensors with 2225 driver sensors. The equate file is required by both translators. Two types of equate files can exist. A specific equate file can be made for each different board. These files have the form "name".EQ with the name and device the same as those designated for the N= network (.SR) file.

If "name".EQ can't be found, a search is made for a file USEREQ.EQ on DSK:. This file can be made and located on DSK: to be used as an equate file for boards with a common interface.

An equate file consists of a series of 2225 EQUATE commands. Refer to paragraph 2.5.2.1 of the 2225 Operation/Test Programming Manual for conventions and considerations concerning EQUATE commands. The EQUATE command relates 2225 driver-sensor pins to test program driver-sensor pins. An equate file is required by both the Image and Test Program Translators. Additionally, this file must equate VCC and GND to 2225 driver-sensor pins. For example:

```
100 EQUATE (103,96), (VCC,107), (GND,108)
```

equates test program driver-sensor 103 to 2225 driver-sensor 96, VCC to 2225 driver-sensor 107, and GND to 2225 driver-sensor 108.

Libraries

Four libraries are used by the Image translator to create an .IM file:

CAPSX.CL	(main CAPS library)
CAPSXA.CL	(overflow to CAPS library)
UPLIBX.CL	(update library)
USERLB.CL	(user library for special elements)

These libraries are necessary for obtaining connectivity data for fault tracing information in the image file.

E.3 2225 MODE OPTIONS

Figure E-2 lists the 2225 Mode options. The options are contained on 2 pages within the mode. Methods of changing page options, advancing pages, etc are described in Section 3. The following paragraphs describe the options available on each page and briefly define their use.

Network File (N=)

This is the name of the .SH file to be used as an input for the Image Translator. To obtain the .SH file, a .SR file is processed through the PREPARE N, T, and G submodes.

Test Program File (T=)

This is the name of the test program file to be used as an input for the Test Program Translator. It is recommended that the .DA file be used as the input file. If no file is specified for T=, the .DA file with the same file name as specified for N= is used. The output from this translator normally has a .T2 extension.

User IC Library File (U=)

The user IC library, if one exists, is the same file that is used in the PREPARE Mode. The user library is described in Section 7.

Listing Device (LI=)

This is the output device on which all hard copy will be printed. The line printer (LPT:) and the Teletype (TTY:) are the only devices supported for this specification.

```
              FILE USAGE

N=DSK:BDNAME.SR   NETWORK FILE
T=DSK:BDNAME.DA   TEST PROGRAM FILE
U=BDNAME.CL       USER IC LIBRARY FILE
LI=TEC:           LISTING DEVICE
W=DSK2:,DSK3:     WORKING DISKS

              EXECUTE SUB MODE (COMMANDS:EY, EN, EO, EA)

MODE    STATUS          DESCRIPTION

 I        Y      IMAGE (NETWORK) TRANSLATOR
 T        Y      TEST PROGRAM (.DA) TRANSLATOR


TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE



SELECT MISCELLANEOUS OPTIONS (COMMANDS SY, SN, SO, SA)      2225 MODE

OPTION   STATUS          DESCRIPTION

  P        Y      PRINT ALL ERRORS ON TELETYPE
  E        Y      INVOKE EDITOR ON ERRORS IN ANY SOURCE FILE
  Q        N      LIST EQUATES FILE
  IM       N      LIST IMAGE (.IM)
  IW       Y      LIST IMAGE TRANSLATION WARNINGS (.IW)
  T2       Y      LIST TRANSLATED TEST PROGRAM (.T2)
  TW       Y      LIST TEST PROGRAM TRANSLATION WARNINGS

TYPE 'HELP' FOR INSTRUCTIONS OR 'EXIT' TO EXIT FROM THIS MODE
```

Figure E-2. 2225 Mode Options.

Working Disks (W=)

The location of the data-base files prepared in the 2225 Mode that are produced for testing using the 2225 are specified by the W= option. Normally, W= is set to reflect the same devices as specified for W= in the PREPARE Mode (refer to paragraph 5.3.2.6). Referring to this paragraph, Table E-1 lists the 2225 Mode output files and the relative working disk where each one is stored.

Table E-1
2225 MODE OUTPUT FILES

Ext.	Name	DSKm:	DSKn:
.IM	Image Mode output		X
.IW	Image Mode Warnings	X	
.T2	Test Program Translator		X
.TW	Test Translator Warnings	X	

Image Network Translator (I)

The 2225 Execute Sub-Mode I switch (Yes/No) selects the Image Translator to be run.

Test Program (.DA) Translator (T)

The 2225 Execute Sub-Mode T switch (Yes/No) selects the Test Program Translator to be run.

Miscellaneous Options

The miscellaneous options are self-explanatory.

E.4 IMAGE TRANSLATOR ERRORS

Errors associated with the Image Mode Translator indicate problems with the equate commands as follows.

SYNTAX ERROR	Self-explanatory
IS NOT A 179X D-S # OR VCC OR GND	Left side of the equate format [(123,456)] must be 1-511 or VCC or GND.
IS NOT A 2225 D-S #	Right-hand side of EQUATE must be within the range 0-382.
2225 D-S # WAS USED MORE THAN ONCE	Two 179X D-S #s were equated to the same 2225 #.
179X D-S # WAS USED MORE THAN ONCE	Two 2225 D-S #s were equated to the same 179X #.
NO EQUATE FOUND	File does not contain an EQUATE statement (perhaps because of an error).
PREMATURE END OF FILE	End of file found before it was expected.

E.5 IMAGE TRANSLATOR WARNINGS

If a successful translation is accomplished a warnings file (.IW) may be present. These are the warnings:

TRANSITIONAL OMITTED	Transitionals can not be translated.
IC WAS NOT FOUND IN LIBRARY	Default connectivity was used for this IC.

NOTE

The following error responses normally mean that some modification must be made to the image file (.IM) after translation.

NO 2225 D-S # FOR THIS 179X D-S #	Equate did not have a 2225 D-S # so this IC was not translated. (Fix the equate and re-translate program.)
NOT PROCESSED DUE TO INPUTS EXCEEDING LINE LENGTH	Maximum line length of 255 characters was exceeded by the inputs on this IC thus, this IC was aborted. Manual intervention is needed.
2225 D-S # FOR VCC OR GND NOT FOUND	The EQUATE file should contain 2225 D-S #'s for VCC and GND if these are used in the circuit. (Make EQUATE commands for these and re-translate.)
IC NOT PROCESSED. NO CONNECTION TO INPUTS OR OUTPUTS	IC was used in .SR, but Image Translator found no connection to either inputs or outputs.

E.6 TEST PROGRAM TRANSLATOR ERRORS

When the .T2 and .TW files are created, the following error messages may result:

DISK FULL	There is not enough room for the resultant .T2 file.
SYNTAX ERROR (USE TRANSLATE MODE)	Re-translate the test program using the CAPS translators, then the 2225 T Translator. The label address of the incorrect statement is also displayed in the form 100:+3 where the failing statement is the third one following the label 100:.

E.7 TEST PROGRAM TRANSLATOR WARNINGS AND MESSAGES

When a test program is translated, the commands that have no 2225 equivalent are placed in the .TW file. All other usable commands are placed in the .T2 output file. Figure E-3 shows an example of a test program and its resulting .T2 and .TW files.

```
/SAMPLE DA  15-MAR-79/

/SAMPLE TEST PROGRAM FOR THE GR TEST-TO-2225 PSP TRANSLATOR!/
10:  ASN LGC IA(1-12) VILA=-0.0 VIHA=5.0
               OA(13,15,17,19,21,23) VOLA=0.5 VOHA=4.5
15:  IH(1-6,9) OL(13,15);
     IL(2) OH(23);
20:  SET FMU AT(1) TRIGA=2.5+;
     MEAS FREQ[15];         /LOOP UNTIL OCS. STARTS/
25:  ASN LGC IA(1,3,7) VILA=-12.0 VIHA=12.0
                                 OA(13-16) VOLA=0.5 VOHA=11.5;
     IH(1,7) OL(13-16);     /CLEAR COUNTERS/
     $IH(3); $IL(3);
     OH(13);
     $IH(3); $IL(3);
     OH(14);
30:  MEAS FREQ;
     $IL( );
     END;
```

.T2 TRANSLATOR OUTPUT FILE

```
5 REM SAMPLE.DA  15-MAR-79
10 REM SAMPLE TEST PROGRAM FOR THE GR TEST-TO-2225 PST TRANSLATOR!
15 IL 1,2,3,4,5,6,7,8,9,10,11,12; OL 13,15,17,19,21,23
20 SET (1,  0.00),(2,  5.00),(3,  2.50)
25 SET (4,  0.00),(5,  0.00),(6,  0.00)
30 L 7,8,10,11,12,13,15,17,19,21,23; H 1,2,3,4,5,6,9; X
35 L 2; H 23; X
40 OX 1,2,3,4,5,6,7,8,9,10,11,12,13,15,17,19,21,23
45 IL 1,3,7; OL 13,14,15,16
50 SET (1,-12.00),(2, 12.00),(3,  6.00)
55 SET (4,  0.00),(5,  0.00),(6,  0.00)
60 REM CLEAR COUNTERS
65 L 3,13,14,15,16; H 1,7; X
70 H 3;
75 L 3;
80 H 13; X
85 H 3;
90 L 3;
95 H 14; X
100 L 1,7;
105 END
```

.TW TRANSLATOR WARNINGS FILE

```
              ***** 2225 TRANSLATOR WARNINGS *****

    179X STATEMENTS NOT TRANSLATED TO 2225 PST LANGUAGE

<>= IF TRANSLATED WOULD HAVE OCCURRED AFTER <2225 LINE #>
[]= OCCURRENCE IN GR TEST PROGRAM WAS AT [STEP #+ # OF STATEMENTS]
TEXT= TEST STATEMENT AS OCCURRED IN GR TEST PROGRAM
**********************************************************************
```

Figure E-3. Sample 2225 Test Program Translation.

E.8 PDP-8 TO 2225 LINK

Any appropriate file such as the .IM and .T2 files can be transferred from a PDP-8 based system to the 2225 using the following procedures. It is assumed that the KL8JA Interface Board has been properly set and installed within the PDP-8. Refer to paragraph E.9. It is also assumed that the user is familiar with 2225 operation.

Perform the following procedure:

a. Power-up the 2225 and PDP-8 based test system. Enter the OS/8 Monitor with the PDP-8.

b. Enter the COMCODE Mode on the 2225.

c. On the Teletype enter: .R 8LINK

d. On the 2225 type: ORG 4800

NOTE

The KL8JA Interface Board must be set for transfers at a 4800 Baud rate.

e. The Teletype responds: PDP-8 SYSTEM READY

The 2225 can now be used to transfer files using the communication commands referenced in paragraph 2.4.5 of the 2225 Operation/Programming Manual.

E.9 KL8JA CONSIDERATIONS

A KL8JA Interface Board located within the PDP-8 provides the hardware for the communications link between the PDP-8 and the 2225. The board has two switches SP1, and SP2 to select both baud rate and I/O addressing. These settings were properly preset when the installation was made, and should not normally have to be set again. The proper switch settings are shown in Figure E-4. Refer to the schematic for the locations of SP1 and SP2.

The device code addresses are 77 (for Transmit) and 10 (for Receive). Note that device code 77 is used by the DEC Tape Unit. If a DEC Tape Unit exists on the system, the I/O board must be removed from the processor. Refer the matter to GenRad if the codes 10 and 77 are used for other purposes.

Switches 1 and 2 of SP1 and SP2 select the baud rate. Figure E-4 shows the configuration for a 4800 Baud rate. Switches 3 through 8 select the device codes as shown.

1	OFF
2	OFF
3	ON
4	ON
5	ON
6	ON
7	ON
8	ON

SP1
Transmit Device Code 77

1	ON
2	ON
3	OFF
4	OFF
5	ON
6	OFF
7	OFF
8	OFF

SP2
Receive Device Code 10

Figure E-4. KL8JA Switch Settings.

INDEX

-- A --

Accept Procedures, B-12
Accepted Part, B-2
ACTIVE, 5-73
Activity Tracking, B-20
Adjacent Pin Shorts Test, 6-48
AFD, 5-80
AFD Results, 5-57
AFD Simulation, 5-17
AFD Status Message, 5-81
AFL Options, 6-58
AFL Simulation, 6-7, 6-60
AFL Sub-Mode, 6-7
AFL Testing, 6-57
AFL to Probe Link, 6-7, 6-61
AFL Transfer Limits, 6-14, 6-34, 6-59, 6-63
AFL/PROBE/FINDS Messages, 6-24
AGN, 5-14
Alias Error Messages, 5-8
Alias Function, 5-7
Alias ICs, 5-61
ALLNODES, 5-43
APG Monitor, 3-2, 3-4, 3-29
APG Use, B-1
APGMTO.BI, 5-89
ARCH, 3-20
Archive, 3-20
Archive Support Program, 3-2, 3-20, 3-26
Arithmetic Relation Operators, C-11
ASP, 3-2, 3-20, 3-26
ASSIGN PNAME, 5-12
Auto Page Advance, 5-99
Automatic Fault Detection, 5-81
Automatic Fault Location, 6-7, 6-12, 6-57
Automatic Fault Simulation, 5-17
AUTOPROGRAM Mode, 3-3
Auxiliary Commands, 3-37

-- B --

BACK Command, B-27
Back Connectivity, B-20
Back-Up Disk, 5-83
BACKUP, 6-44
BATCH Mode, 3-3
BATCH Processing, 3-43
Bed-of-Nails, 6-5
BEGIN/FINISH Statement (LOG), C-10
Best Fit Location, 3-12
BLACKBOX Coding, 2-31
BLACKBOX Model, 2-31
BLOB Element, 2-29
Boot Elements, 7-28
BUILD Program, 3-9
Building Blocks, 7-16
Bus Contention, 5-39

-- C --

CAPMTO.BI, 5-89
CAPS Data Base Files, A-1
CAPS Libraries, 7-1
CAPS Monitor, 3-2, 3-4, 3-29
CAPS Translator, 5-12
CAPSX.CL, 5-82
Card Reader, 3-9
Cartridge (Back-Up), 5-83
CASE UNTIL, B-18
CCDATA.lb, 5-32
CCL, 3-1
CCL Commands, 3-6, 3-10
Check Error, 2-12
Check Mode, 2-10, 2-11
Checkpointing, 6-18, 6-36
Circuit Coding, 2-25
Circuit Description, 2-24, 2-25
Circuit Description Analysis, 5-4
CLEANUP, 5-83
CLEANUP Mode, 5-98
CLEANUP Program, 3-3
CLIP Command, 2-18
CLIPD Command, 2-23
Closed Loop, 2-3
CM File, 5-17
Coding, 2-25
Coding a Circuit Description, 2-24
Coding Guidelines, 2-13
Coding the Network Description, 2-2
Command Decoder, 3-8
Compare File, 5-17
Computer-Guided Probe/Clip Testing, 6-33
Concise Command Language, 3-1
Conditional Connectivity File, 5-23, 5-32
Conditional Connectivity Data Base, 5-2
Conditional Connectivity, 6-16
Conditionality, 6-36
Connection Error, 2-12
Connection Mode, 2-10, 2-11
CONNECTIONS Field, 2-5
CONVERT Program, 3-3
Copy (.COP), 3-7
CORE ALLOCATION, 5-63
Cross-Reference Data, 5-9, 5-66
CTRAN, 5-12

-- D --

D Sub-Mode, 5-2, 5-15, 5-80
DA File, 5-1, 5-14, 5-15, 5-69, 5-70
Data Logging, C-1
Data Terminal Display, 3-9
DATE Command, 3-5
Default Device, 3-8
DEFINE (Macro), 5-11
Delay Element, 2-27
DELETE (.DEL), 3-6
Destructive File Transfer, 3-11, 3-13, 3-16
Device Adaptor, 2-1
Device Adaptor Preparation, 2-1
Device Adaptor Wiring, 2-24
DF File, 5-14, 5-70
DIAGNOSE Mode, 1-2, 1-4, 3-3, 6-1
DIAGNOSE Mode Options, 6-11
DIAGNOSIS CONTROL BY CAPS SECTION, 6-10, 6-12
Diagnosis Mode Preparation, 5-2, 5-15
Diagnostic Error Messages, 6-23
DIAGNOSTIC MODES PAGE, 6-12
Diagnostic Resolution Module, 2-1, 6-5
Difference Report File, 5-14
Digital Driver/Sensor Display, 6-29
DIRectory (.DIR), 3-6
Directory Order, 3-3
Disk (Back-Up), 5-83
Disk (Testing Only), 5-86
Disk Cartridges, 1-5
Disk Drive, 3-8
DISPLAY, 3-4
DISPLAY MODE, 5-94
Display Options, 5-61
Double-Cross Check, 2-3
Driver/Sensor Pin Coding, 2-13
DRM, 2-1, 6-5
DRM (Intermediate), 6-8
DRM Sensor Threshold Selection, 2-33

-- E --

E/T/C Mode, 1-3, 5-9, 5-80, B-20
E/T/C Mode Commands, B-24
EDITEDIT on Error, 5-60
EDIT Program, 3-2
Empty File, 3-12
EQ File, E-2
Equate Command, 3-36
Equate File, E-2
EQUATION Command, B-27

Equation Generation, Tracking,
  and Connectivity, 1-3, B-20
Equation Generation Output, B-35
Equation Generation, B-20
Equipment Shut-Down, 4-5
Error Messages (Archive), 3-24
Error Messages (Alias), 5-8
Error Messages (Conditional Connectivity), 5-39
Error Messages (Macro), 5-10
Error Messages (Pre-Processor), 5-4, 5-6
Error Recovery, 6-2, 6-55
Errors (Diagnostic), 6-23
Errors (Image Translator), E-5
Errors (UD Translation), 5-45
ET File, 5-9, 5-68
ETC Mode, 3-4
EXECUTE SUB-MODE, 5-52
Expanded Test Program, 5-68
Extended Track Short Detection, 6-18
EXTernal Coding, 2-13
EXTernal Name, 2-3
EXTernal Signal Path, 2-3, 2-13
External Signature, 5-19

-- F --

F Sub-Mode, 5-2, 5-17, 5-73, 5-81
FAKE Attribute, 7-5
FAKE Command, 2-18
FAKE Element, 5-43
FAMILY Command, 2-18
Fault Classes, 5-18, 5-73
Fault Collapsing, 5-19
Fault Detection, 5-17
Fault Detection Simulation, 5-2
Fault Dictionary, 5-17, 5-19, 6-7
Fault Dictionary File, 5-20
Fault Group, 5-19
Fault Identifying Nodal Diagnostic Software, 6-5
Fault Modeling Options, 5-55
Fault Simulation, 1-3, 5-18
FAULT SIMULATION STATUS, 5-57
FG File, 5-73, 5-81
File, 3-12
File Initialization, C-6
File Management, 3-10
File Oriented Transfer Program, 3-1
File Structured Device, 3-8
FILE USAGE (DIAGNOSE), 6-11
File-Structured Device, 3-12
FINDS, 6-5, 6-56
FINDS Display, 6-49
FINDS IC Testing, 6-51
FINDS Microvoltmeter Option, 6-18
FINDS Shorts Test, 6-50
FINDS Status Message, 6-50
FINDS Testing, 6-47
FLFORM Program, A-5
Floppy Diskette, 3-9, A-4
Format (Disk), 5-84
FORMAT EQUATION, B-27
FORMAT FINAL, B-25
FORMAT PATH, B-25
FORMAT UPDATE, B-25
FOTP, 3-1
FP File, 5-75
FX File, 5-20

-- G --

G Sub-Mode, 5-1, 5-14, 5-68, 5-80
GENERATE Mode, 1-3, 1-4, 3-4, 7-1, 7-9
GenRad Monitor, 3-1, 3-29
GND Element, 2-29
Good Board Simulation, 5-70
Good Network Simulation, 5-1, 5-14, 5-68
Guided Probe History, 6-46
GUIDED PROBE OPTIONS, 6-15
Guided Probe/Clip, 6-2, 6-36
Guided Probe/Clip Message, 6-38

-- H --

HDS Monitor, 3-5
HELP CCL, 3-7
HELP Command, 3-37

-- I --

I Sub-Mode, 5-48, B-2
I7 File, 5-39
IC Building Blocks, 7-28
IC Bus Pins, 7-20
IC Clip, 2-2, 6-4, 6-17, D-3
IC Library, 1-4, 7-1
IC Modeling, 7-1, 7-16
ICDELETE, 3-4, 7-14
ICINFO, 3-4, 7-14, 7-20
IF Statement (LOG), C-10
Image Translator, E-1
Image Translator Errors, E-5
Image Translator Warnings, E-5
Inc-Sim Options, 5-63
Incremental Diagnosis, B-15
Incremental Simulation, 1-3, B-1
Incremental Simulation Options, B-2
Incremental TP Preparation, 5-2, 5-9, 5-48
Initial AFL Lookup, 6-59
Initiate Process Commands, B-27
Input/Output Coding, 2-13
Interactive Teletype Keys, 5-100
Intermediate DRM, 6-8, 6-62
Intermediate Name, 2-3, 2-9
Intra-Mode Command, 3-35

-- J --

Jumping-Point Data Base, 5-2
Jumping-Point Error Messages, 5-29
Jumping-Point File, 5-23, 5-29, 6-16
Jumping-Point Letter Commands, 5-31

-- K --

Keyboard Monitor, 3-1, 3-5
KL8JA Interface, E-8

-- L --

Largest Empty Location, 3-12
LIBLIST, 3-4, 7-11
Library, 5-82
Library Element Deletion, 7-14
Library Element Information, 7-14
Library Listings, 7-11
Library Utility Programs, 7-11
Line Printer, 3-9
LIST, 3-7
LISTALL, 3-4, 5-81, 5-91
LOCATION Message Command, 2-19
Log Data Collection, C-7
Log File Report Generator, C-8
Log File Status Display, C-7
LOG Mode, 3-4, C-1
LOG Mode Printout, C-20
LOGGING Mode Options, C-4
LOGGING OPTIONS Page, 6-18
Logic Timing Considerations, 7-16
Long Error Messages, 5-60
LOOK, 3-7, 5-93
LOOK Program, 3-3
LOOP/STOPLOOP, B-17

-- M --

Macro Error Messages, 5-10
Macro File, 5-50
Macro Processor, 5-9, 5-60
Magnetic Tape, 3-8
MAKE 1795 Files, A-2

INDEX (Cont.)

MAKE DISK Options, A-3
MAKE Mode, 3-4, A-1
MANUAL Sub-Mode, 6-10, 6-29, 6-64
MEMORY, 3-7
Merging Loop, 2-10
MESSAGE Command, 2-20
Messages (Long Error), 5-60
Microvoltage Test, 6-48
Multiple Attributes, 7-6
Multiple CAPS, 6-10

-- N --

N Sub-Mode, 5-1, 5-4, 5-63
NAME Field, 2-4
Network Description Analysis, 5-1, 5-4
Network Description, 2-2, 2-24, 2-25, E-1
NETwork Mode, 3-3
NOCLIP Command, 2-15
NODAL DATA, 5-43
Nodal Verification, 3-5, 5-25, D-1
NODALV, 3-5, 5-25, D-1
NODALV Options, D-2
NOFAULT Command, 2-22
NOFUN, 2-22, 7-6
Non-Destructive File Transfer, 3-11, 3-15, 3-17
Non-Digital Display, 6-31
Non-File Structured Device, 3-9, 3-12
Non-Recursive Building Block, 7-17
NONAIL Command, 2-22
NOOHM Command, 2-23
NOPRINT Command, 2-20
NOPROBE, 2-21, 7-6
NOSHORT Command, 2-22
NOUV Command, 2-23
Numeric Selection Variables, C-11
NVTRAN, 5-25, D-8

-- O --

ODT Program, 5-86
One-Shot Element, 2-28
ONESHT, 2-28
OPERATOR CONTROLS Page, 6-21
Option ON/OFF Command, 3-36
OS/8 Batch Mode, 3-45, 5-89
OS/8 Commands, 3-6, 3-44, 3-46
OS/8 Monitor, 3-1, 3-5

-- P --

P Sub-Mode, 5-2, 5-23, 5-80
Paper Tape Punch, 3-9
Paper Tape Reader, 3-9
Passive Probe, 2-2
PD File, 5-80
Percent Processing, 2-14
Percent Processing Commands, 2-15, 2-16
Percent Processing Updates, 5-2, 5-45
Peripheral Interchange Program, 3-1
Permanent File, 3-12
Pin Activation Results, 5-73
Pin Loading Data, 7-7
PIP, 3-1
PN File, 5-12
Positional Argument, 3-38
POWER Command, 2-21
Power Short Test, 6-48
PP Batch, B-11
Pre-Processor, 2-9, 5-4
Pre-Processor Error Messages, 5-6
PREPARE Mode, 1-2, 1-3, 3-4, 5-1
PREPARE Mode Data Output, 5-63
PREPARE Mode Display Options, 5-62
PREPARE Mode Files, 5-51
PREPARE Mode Options, 5-48
PRETTY File, 5-3, 5-70, 5-83
PRINT Command, 2-20
PRINT DIAGNOSIS, 6-59
PRINT OPTIONS Page, 6-13
PRINT Statement (LOG), C-8
Probe (Passive), 6-3
Probe (Pulse-Catching, 6-3, 6-41, D-3
Probe (X1), D-3
PROBE Command, 2-21
Probe Display, 6-32
Probe File, 5-2, 5-23
Probe History, 6-46
Probe Link, 6-33
Probe Sensor Pin Coding, 2-14
Probe Training File, 5-27
Probe/Clip Preparation, 2-2
Probe/Clip Results Code, 6-47
Probe/Clip Testing, 6-33
PROBED Command, 2-23
PULLUP Element, 2-29
Pulse Discrimination File, 5-23, 5-40, 6-16, D-1
Pulse Element, 2-27
Pulse-Catching Probe, 2-1, 2-2, 6-3, 6-41, D-3

-- Q --

Q Sub-Mode, 5-2, 5-45
QL File, 5-2, 5-45
Quick Dictionary Lookup, 5-82, 6-5, 6-33
Quick Lookup, 5-2, 5-45, 6-34
QUIT Program, 4-5
QUIT Statement (LOG), C-10

-- R --

R Command, 3-6
Re-Start, 4-5
Recursive Building Blocks, 7-16
Reject Procedures, B-13
REMember Program, 3-47
RENAME (.REN), 3-7
Rename Alias ICs, 5-61
Renaming IC Devices, 5-7
REPEAT n UNTIL, B-15
REPROBE, 6-44
RESET Element, 2-27
Resistor Selection, DRM I, 2-33
RG File, 7-21
RK8E Disk Assignments, 5-85
RKFORM, 5-83
ROM Generation, 7-21, 7-25
ROM Input File, 7-22

-- S --

SAVE Command, 3-38, 3-41
SAVE File, 5-88
SELECT MISCELLANEOUS OPTIONS, 5-58, 6-19
Selected Fault Command, 5-22
Selection Variables, C-11
Selective Fault File, 5-18
Selective Faulting, 5-20, 5-57, 5-78, 5-82
Sensor Pin Coding, Probe, 2-14
SETUP Program, 3-3
SF File, 5-18
SH File, E-1
Shorts Check Mode, D-10
Signal Path, 2-3
Single Step Expansion, 5-62
Software Required, 4-1
Special Purpose Element, 2-27
Special Simulation Package, 7-2
SQUISH (.SQ), 3-6
SQUISH Command, 3-11, 3-14
SR Analysis, 5-1, 5-4
SR File, 2-1, 5-1, 5-63, 7-2
SSP, 7-2
SSP-BOOT Element, 7-17
ST File, 5-61
Starting PDP8/A Systems, 4-2
Starting PDP8/E Systems, 4-3
Starting-Point Data Base, 5-2
Starting-Point Error Messages, 5-29
Starting-Point File, 5-23, 5-26, 6-15
Starting-Point Letter Commands, 5-28

INDEX (Cont.)

Statement Field, 2-3
SUBMIT Command, 3-46
Suspected Track Short, 6-54
SV File, 5-88
System Device, 3-8
System ShutDown, 4-5
System Software, 3-1, 4-1
System Start-Up, 4-1

-- T --

T Sub-Mode, 5-1, 5-9, 5-67, 5-80
TA File, 5-80
TD File, 5-80
TEC Pages, 5-99
Teletype, 3-9
Termination Identifiers, B-35
Test Generation Language, B-15
TEST Monitor, 3-2, 3-29
Test Program Translation, 5-1, 5-9
Test Program Translator Warnings (2225), E-6
Test Program Translator Errors (2225), E-6
Test Program Translator (2225), E-2
Test Program Translator, 5-12
Testing Only Cartridge, 5-86
TGL, B-15
TGL Statements, B-19
TH Command, 2-32
THRESHold Command, 2-21
Threshold Selection, DRM II, 2-36
Threshold Specification, DRM I, 2-33
TL Command, 2-32
TL File, 5-80
TP File, 2-1, 5-70, 6-11
TP Translation, 5-1, 5-9
TRACK Command, B-27
Track Short Test, 6-48
Tracking Display, B-34
Transitional Commands, 2-32
Transitional Identifiers, B-35
TRANSLATE Program, 3-3
Translator Errors, 5-13
Translator Errors (UD), 5-45
Trial Part, B-2
TRY Batch, B-9
TRYD Batch, B-12
TRYF Batch, B-8
TRYNE Batch, B-10
TRYNW Batch, B-9
TRYSF Batch, B-11
TTY Keys, 5-100
TV File, 5-80
TX Command, 2-32
TYPE Field, 2-5

-- U --

UC File, 5-2, 5-23, 5-32
UD File, 5-2, 5-23, 5-40, D-1
UJ File, 5-2, 5-23
Update Library, 7-1, 7-9
UPDATE Program, 5-86
UR File, 6-11
US File, 5-2, 5-23, 5-26
User Discrimination Data Base, 5-2
User Interaction Sub-Mode, D-7
User Library, 1-4, 7-9
User Macro File, 5-50
User Options, 6-21
User Subroutine File, 6-11
User-Language Macro, 3-25
User-Styled Macro, 5-10
USERLB.CL, 5-82
Utility Programs, 5-83
UV Command, 2-23

-- V --

VCC Element, 2-29
VERIFY (.VERI), 3-7
VERIFY Command, 3-10, 3-14
Verifying Peripheral Interchange
  Program, 3-1, 3-11, 3-13
VPIP, 3-1, 3-11, 3-13
VPIP Error Message, 3-18
VPIP Options, 3-15

-- W --

Warnings (Image Translator), E-5
WIRE Element, 2-30
WN File, 5-63, 5-65

-- X --

X-1 Probe, 2-1, 2-2, 2-14, 6-3, D-3
XR File, 5-9, 5-66

-- Z --

Z Sub-Mode, 5-2, 5-45
ZERO (.ZER), 3-7
ZERO Command, 3-11, 3-14
ZRODSK, 5-83

-- NUMERIC --

2225 Link, E-7
2225 Mode, 3-5, E-1
2225 Mode Options, E-3

NOTES

GenRad

Production Test Division
300 Baker Avenue
Concord, Massachusetts 01742

USA

***Boston** • (Sales) Waltham, MA 02254
Tel: (617) 890-4900 • TWX: 710 324 0893
(Service) Concord, MA 01742
Tel: (617) 369-4400 • TWX: 710 347 1051

***Chicago** • 1156 Shure Drive, Arlington Heights, IL 60004
Tel: (312) 577-1881 • TWX: 910 291 1209

***Cincinnati** • Suite A, 2002 Ford Circle, Milford, OH 45150
Tel: (513) 831-9210 • TWX: 810 460 8323

***Colorado Springs** • 6180 Lehman Drive, Suite B207, Colorado Springs, CO 80907
Tel: (303) 593-8238

***Dallas** • 1121 Rockingham Lane, Suite 100, Richardson, TX 75080
Tel: (214) 234-3357 • TWX: 910 867 4771

***Los Angeles** • P.O. Box 19500, 17361 Armstrong Avenue,
Irvine Industrial Complex, Irvine, CA 92714
Tel: (714) 540-9830 • TWX: 910 595 1762

***New York** • 22-08 Route 208, Fair Lawn, NJ 07410
Tel: (NJ) (201) 797-8001; (NY) (212) 964-2722 • TWX: 710 988 2205

***Orlando** • 3751 Maguire Blvd., Suite 170, Orlando, Florida 32803
Tel: (305) 894-4303 • TWX: 810 850 0270

***San Francisco** • 2855 Bowers Avenue, Santa Clara, CA 95051
Tel: (408) 727-4400 • TWX: 910 338 0291

***Washington, D.C.** • 1701 Research Boulevard, Rockville, MD 20850
Tel: (301) 424-6224 • TWX: 710 828 9783

CANADA

***Toronto** • 307 Evans Avenue, Toronto, Ontario M8Z 1K2
Tel: (416) 252-3395 • TLX: 06-967624
Montreal • Tel: (514) 747-1052 • TLX: 05-826652

EUROPE

***England** • Norreys Drive, Maidenhead, Berkshire SL6 4BP England
Tel: (0628) 39181 • TLX: 851-848321
Best • Tel: (0) 49984240 • TLX: 844-59535
***Milan** • Tel: (02) 8466541 • TLX: 843-320373
***Munich** • Tel: (089) 41690 • TLX: 841-529917
***Paris** • Tel: (01) 7970739 • TLX: 842-220991
Rennes • Tel: (099) 535154
***Rome** • Tel: (06) 455839
Wiesbaden • Tel: (06121) 370057 • TLX: 841-04186489
***Zurich** • Tel: (01) 552420 • TLX: 845-53638

ASIA

Japan • Tokyo Electron Limited, Shinjuku-Nomura Bldg.,
1-26-2 Nishi-Shinjuku, Shinjuku-Ku, Tokyo 160, Japan
Tel: 343-4411 • TLX: 232-2240 Labtel J
Other countries in Asia and Pacific • Tel: 344-9245 (Tokyo)

LATIN AMERICA

Contact WALTHAM, MA 02254 USA
Cable GENRAD Waltham (MASS.)

*Service Facilities Available

Printed in USA